Mensile - 25 novembre 1985 - Anno IV - N. 25 - Sped. Abb. Post. Gr. III/70 CR - Distr. MePe

Speciale simulazioni

Automa cellulare

Simula robot

Variazioni demografiche

Effetti sonori

La grande enciclopedia delle routine

Gestione del mazzo di carte

Sommario

RUBRICHE

Direttore: Alessandro de Simone
Redazione/collaboratori: Giovanni Bellù, Simone Bettola, Andrea e Alberto Boriani, Giancarlo Castanga, Eugenio Coppari, Maurizio Dell'Abate, Marco De Martino, Luca Galluzzi, Giancarlo Mariani, Flavio Molinari, Enrico Scelsa, D. Matturro, M.L. Nitti, Massimo Patti, Massimo Pollutri, Carla Rampi, Fabio Sorgato, Giovanni Verrelli, Antonio Visconti.
Segreteria di redazione: Maura Ceccaroli, Piera Perin
Ufficio Grafico: Mary Benvenuto, Arturo Ciaglia, Paolo Vertuccio
Direzione, redazione, pubblicità: V.le Famagosta. 75 - 20142 Milano - Tel. 02/8467348
Pubblicità: Milano: Mirco Croce (coordinatore), Michela Prandini, Giorgio Ruffoni, Roberto Sghirinzetti, Claudio Tidone, Villa Claudio - Segretaria: Lilliana Degiorgi
● Roma: Spazio Nuovo - via P. Foscari 70 - 00139 Roma - Tel. 06/8109679
Abbonamenti: Marina Vantini
Tariffe: prezzo per copia L. 3.000. Abbonamento annuo (11 fascicoli) L. 28.000. Estero: il doppio. Abbonamento cumulativo alle riviste Computer e Commodore Computer Club L. 55.000.
I versamenti vanno indirizzati a: Systems Editoriale Srl mediante assegno bancario o utilizzando il c/c postale n. 37952207
Composizioni: Systems Editoriale Srl
Fotolito: Systems Editoriale Srl
Stampa: La Litografica S.r.l. - Busto Arsizio (VA)
Registrazione: Tribunale di Milano n. 370 del 2/10/82 - Direttore Responsabile: Michele Di Pisa
Sped. in abb. post. gr. III - Pubblicità inferiore al 70%
Distribuzione: MePe, via G. Carcano 32 - Milano

Simulare, che passione!

Quando la ricerca scientifica, a causa della complessità sempre crescente degli esperimenti, si è scontrata con la dura realtà delle spese per sostenerli, si è cercato di far ricorso ad una metodologia che potesse, per quanto possibile, evitare spreco di tempo, mezzi e materiali.

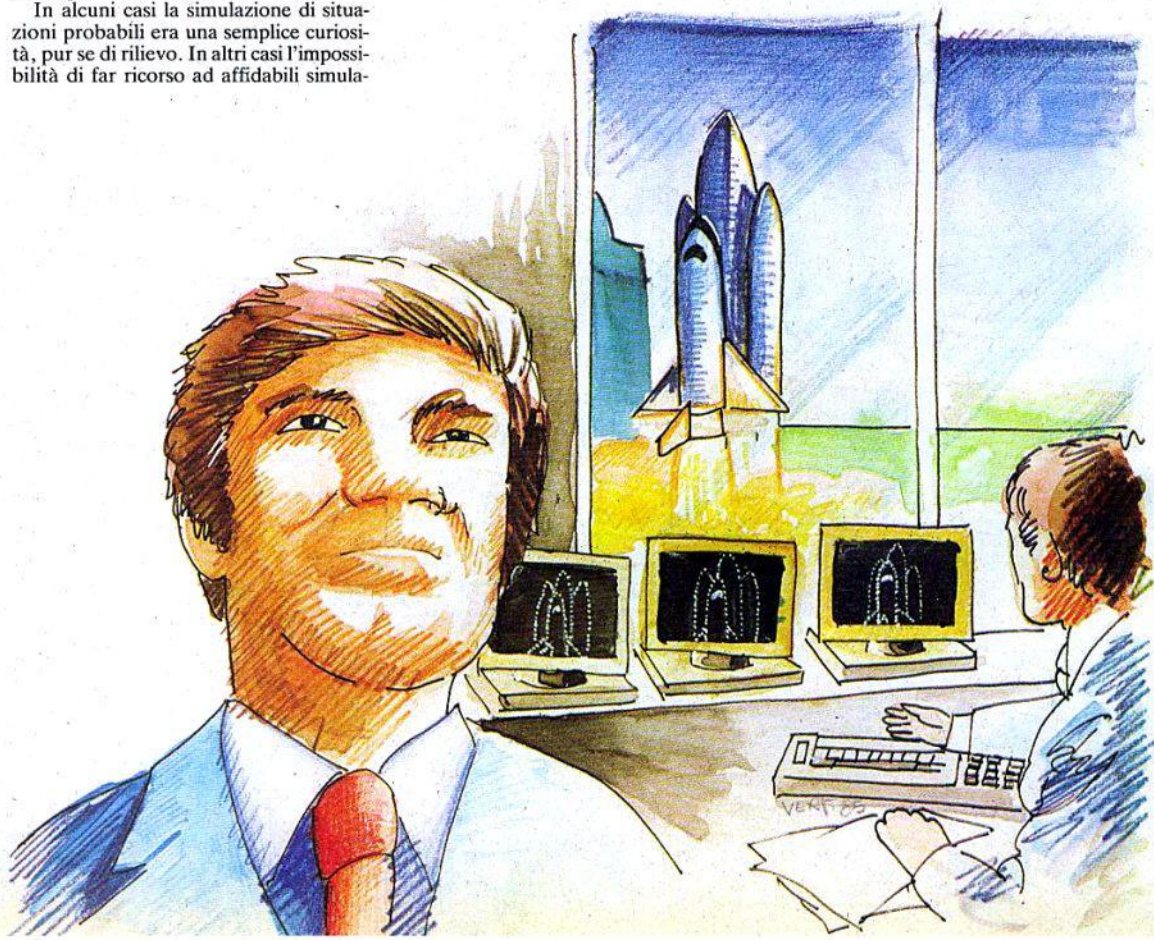
D'altra parte, la crescente ansia di offrire ad un pubblico sempre più esigente prodotti competitivi spingeva le grandi industrie a favorire analoghe ricerche che consentissero un risparmio sensibile sulla sperimentazione di prototipi.

In alcuni casi la simulazione di situazioni probabili era una semplice curiosità, pur se di rilievo. In altri casi l'impossibilità di far ricorso ad affidabili simulazioni ha impedito lo svolgersi di taluni eventi che la tecnologia avrebbe sì consentito, ma con un margine di rischio inaccettabile.

Consideriamo, ad esempio, le imprese spaziali che, da una parte, richiedevano l'impiego di ingenti capitali e, dall'altra, la necessità di rassicurare l'opinione pubblica circa la certezza del rientro sulla Terra dell'equipaggio umano.

Ancor oggi, come negli anni '60, prima di premere il pulsante che abilita la partenza del vettore, vengono effettuate migliaia di simulazioni basate su modelli matematici che tengono conto di qualsiasi evento e della influenza che può avere nella riuscita dell'operazione.

Non è un mistero che agenti della C.I.A. e dell'F.B.I. seguono la vita privata degli astronauti e riferiscono, a chi di dovere, la cronaca di ciascuno di essi. Litigi in famiglia, contrarietà di ordine fisico o psicologico potrebbero, infatti, influire negativamente sul comporta-

mento durante la missione e pregiudicarne l'esito finale.

Vengono attivati, pertanto, modelli matematici, vale a dire complesse procedure logico-deduttive che tengono conto delle reazioni che è probabile si manifestino sia negli uomini sia nelle "cose". In queste formule compaiono, opportunamente quantificati, i valori relativi agli aspetti meteorologici come a quelli delle qualità e quantità di carburante; all'affidabilità dei materiali che costituiscono le tute o i cristalli corazzati; alle rotte degli aerei che saranno presenti al momento del lancio oppure la probabilità di incontrare stormi di uccelli migratori durante il rientro; al dubbio che l'attenzione di un astronauta sia distolta da preoccupazioni circa la salute di un congiunto eccetera.

Le formule di cui stiamo parlando sono applicabili talvolta con rigore scientifico, ma in certi casi devono basarsi su esperienze precedenti o semplicemente empiriche (come i bioritmi).

Di certo è che le circostanze che contribuiscono alla buona riuscita di un'impresa spaziale sono talmente numerose che risulta indispensabile affidarsi ad una intera rete di elaborazione in cui più computer siano collegati tra loro in una sinergia incredibilmente rapida quanto efficiente.

E' noto infatti che, in molti casi, il segnale di allarme di uno solo di questi computer ha spesso bloccato il conto alla rovescia perchè la sicurezza di una sola delle migliaia di operazioni eseguite aveva valutato inaccettabile una semplice probabilità di imprecisione.

Le simulazioni, ovviamente, non necessariamente devono essere eseguite da grossi computer (i cosiddetti mainfraine) ma, con un po' di buona volontà, anche dal piccolo home computer che ognuno di noi possiede. E' sufficiente, infatti, non pretendere la luna da un calcolatore che è nato per altri scopi (e con altri limiti) e la soddisfazione è garantita!

Se riflettiamo, d'altronde, anche un videogioco può essere considerato un modello matematico e, a volte, una vera e propria simulazione.

Pensiamo ad una delle numerose versioni di corse automobilistiche in cui è necessario mantenere la propria vettura in carreggiata, evitare il guard-rail o le auto concorrenti, stimare l'angolo della curva e decidere se la velocità con cui viaggiamo sia eccessiva o meno.

Passando agli adventure non possiamo fare a meno di notare il realismo con cui

essi sono progettati e la corrispondenza alla realtà nel caso, pur se improbabile, si dovessero verificare determinate situazioni.

Anche i simulatori di volo, che in alcune versioni sono dei veri e propri gioielli di mini-informatica, ci hanno spesso meravigliato per la loro incredibile aderenza alla realtà.

Dato che siamo in tema, parliamo ora di un "vero" simulatore, unico al mondo, inaugurato di recente dalla Mercedes - Benz e presentato sul numero di luglio della rivista "Quattroruote".

Pensate che hanno dovuto costruire un intero edificio allo scopo di alloggiare le numerose apparecchiature che hanno il compito, simulando automobili che sono ancora "sulla carta", di studiare il loro comportamento reale e decidere se iniziarne la produzione o meno.

E' inutile soffermarsi sulla utilità economica di simili iniziative!

Lo "sperimentatore" entra in una grande stanza in cui è presente la carrozzeria del prototipo in questione. Attraverso il parabrezza, i finestrini laterali ed il lunotto si possono a stento notare numerosi schermi giganti, ben dissimulati, che circondano il prototipo stesso.

Come avrete intuito, sullo schermo vengono proiettate immagini a colori in alta risoluzione generate da una batteria di computer che riproducono, in pratica, un vero e proprio "ambiente" in cui si troverebbe il guidatore se guidasse quell'automobile. Tutti i comandi di questa sono collegati ad altrettanti computer; per cui, se il guidatore gira il volante, il paesaggio "cambia" proprio come se la manovra fosse effettuata; se accelera, il rumore del motore aumenta, la prospettiva cambia e, se è il caso, si sente il rumore della sgommata tipica dell'eccessiva accelerazione. Ma non basta: in quest'ultimo caso il muso della vettura si inclina in alto grazie ad alcuni meccanismi che riproducono l'effetto dell'accelerazione improvvisa. In caso di brusche frenate la macchina, se è il caso, sbanda e se in quel momento sopraggiungeva un'auto (simulata sullo schermo del lunotto) si avverte non solo il rumore ma addirittura il sobbalzo dello schianto che, assicurano, è del tutto realistico e proporzionale all'entità del danno!.

La perfetta aderenza alla realtà è tale che, dopo un po' di tempo, è praticamente impossibile rendersi conto che l'auto che si guida è ferma, e che il movimento è solo opera di potenti computers che lavorano insieme alla velocità di milioni di operazioni al secondo.

Alessandro de Simone

Istruzioni di salto L.M.

□ **Che significato ha il valore che si trova dopo un'istruzione di salto (BNE, BEQ, eccetera) nel Linguaggio Macchina?** (Stefano Rosa - Venezia)

• C'è da precisare, anzitutto, che il salto viene eseguito a patto che la condizione si verifichi, altrimenti viene ignorata ed il programma prosegue con l'istruzione successiva.

Nel caso in cui il salto debba essere eseguito, se il numero che segue l'istruzione di salto è inferiore a 128, il programma proseguirà "saltando" tante locazioni di memoria quante ne indica il valore stesso.
Se il numero è invece superiore a 127 (ha, in altre parole, il settimo bit settato), il salto va effettuato facendo il conteggio all'indietro delle locazioni da saltare "alle spalle" dell'istruzione stessa. Ad esempio, considera il seguente segmento di programma L.M.:

```
0344 A9 FF
0346 8D 00 80
0349 EE 47 03
034C D0 F6
034E AD 48 03
```

Quando il contatore di programma si appresterà ad eseguire l'istruzione D0 (indirizzo 034C), terrà conto del risultato dell'incremento del contenuto della locazione effettuata (vedi EE 47 03). Se questo porta un risultato uguale a zero, l'istruzione D0 viene ignorata e, di conseguenza, viene eseguita la AD 48 03.

Se invece il risultato è uguale a zero, il contatore di programma si posizionerà all'indietro (dato che il settimo bit di F6 "esiste"). La locazione 034E, successiva all'istruzione D0 F6, viene denominata convenzionalmente FF, ai fini del conteggio, e, contando all'indietro, si terrà conto del seguente procedimento:

```
.......
F4 0343
F5 0344
F6 0345
F7 0346
F8 0347
F9 0348
FA 0349
FB 034A
FC 034B
FD 034C
FE 034D
FF 034E
.......
```

Dato che l'argomento di D0 è F6, l'istruzione che sarà eseguita sarà quella contenuta in 0344 (A9 FF). Attento, dunque, a contare con cura il "salto" altrimenti rischi di far elaborare istruzioni indesiderate o "a metà"!!

Rubar gratis programmi

□ **Si può trasformare il telefono di casa in modem? E' possibile rubare i dati di un altro personal servendosi di un modem? L'uso del telefono è gratuito o gli scatti vanno pagati?** (Antonio Raineri - Melegnano)

• Il modem è un apparecchio che collega un computer, tramite il telefono, con una Banca Dati. Proprio per evitare che qualcuno utilizzi indebitamente (=rubi) il servizio offerto, sono numerosi gli accorgimenti presi per evitare l'inconveniente citato.

Gli avvenimenti del film "War Games", ed i recenti furti di denaro grazie a tessere magnetiche false, devono ritenersi poco probabili e di sempre più difficile realizzazione.

Il collegamento telefonico è possibile, in Italia, pagando un canone aggiuntivo alla SIP, che deve essere in ogni caso informata dell'utilizzo inconsueto che si intende fare del telefono.

Per rubar programmi via telefono è necessario, purtroppo, pagare gli scatti, che vengono addebitati come se si effettuasse una normale teleselezione (magari internazionale). Spero che questo particolare ti faccia desistere da tentativi illeciti di entrare, pur se elettronicamente, negli archivi delle Banche Dati!

Grafica su stampanti

□ **Le stampanti Commodore sono grafiche?** (Massimo Proia - Piacenza)

• I caratteri delle stampanti MPS 801, MPS 802 e MPS 803 sono riprodotti in una matrice di 8x8 punti elementari. Ciò significa che, analogamente a quanto avviene per i caratteri visualizzati sullo schermo, file di puntini, presenti oppure assenti, consentono l'individuazione di un carattere numerico, alfabetico o grafico che sia.

Le stampanti citate, in modo del tutto analogo al computer, permettono all'utente di definire caratteri ricorrendo alla solita tecnica di assegnare un byte a ciascuna fila di otto punti. Sul libretto di istruzioni delle stampanti sono riportati, comunque, alcuni esempi pratici che illustrano piuttosto bene il modo di programmare i caratteri.

Caratteri illegibili

□ **D/ Vorrei sapere se è normale che, scrivendo in rosso su alcuni colori di sfondo dello schermo, i caratteri siano illegibili.** (Alfonso Forgione - Gesualdo)

• Purtroppo non tutte le combinazioni di colori fondo - carattere sono soddisfacenti ed il motivo è da attribuirsi alle "sbavature" (perdonate la terminologia approssimativa) di un colore nell'altro. Proprio il rosso, per esempio, è illeggibile su fondo blu come pure il blu su fondo rosso.

Un'accurata regolazione del contrasto e del colore sul monitor (o televisore) riesce tuttavia, a seconda delle combinazioni impostate, a limitare l'inconveninete lamentato.

Pulsante di reset

□ **Come si può costruire un pulsante per resettare il computer quando si blocca?** (Beppe Almansi - Brescia)

• Il pulsante di reset è un pulsante che, mettendo a massa il piedino RST (Reset) del microprocessore, lo obbliga ad eseguire la procedura di inizializzazione che si verifica tutte le volte che si accende il calcolatore.

In questo modo, però, i puntatori dell'eventuale programma BASIC vengono alterati, come pure rischiano di essere cancellati considerevoli "segmenti" di programma in linguaggio macchina precedentemente allocati.

La connessione di un pulsante (normalmente aperto) tra massa e pin RST della porta utente (o di espansione), deve comunque essere effettuato da personale esperto in montaggi elettronici al fine di evitare pericolosi corti circuiti involontari.

Tabelle

□ **Vorrei sapere come si fa a programmare una tabella in cui in ascisse appaiono i mesi dell'anno e in ordinate i giorni.** (Lino De Togni - Arese)

• Come puoi notare, da un po' di tempo Commodore Computer Club insegna a fare molte "cose", tra cui giochi, utility, tabelle. Il modo in cui gli argomenti sono affrontati è decisamente semplice, tanto da esser digerito da un'utenza che sia anche alle prime armi.

Il programma richiesto, se provi a studiare i listati pubblicati dalla nostra rivista, dovresti essere in grado di farlo da solo.

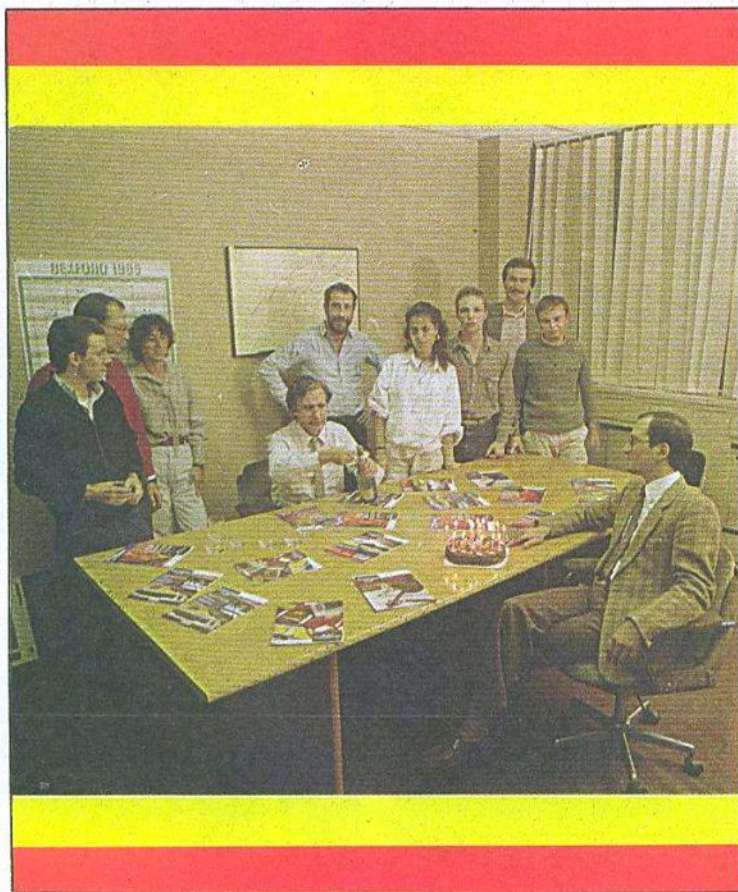
AVVERTENZA IMPORTANTE

Tra le numerose domande che pervengono in Redazione, vengono scelte in prevalenza quelle di interesse generale, rinviando, spesso sine die, la pubblicazione di quelle ritenute meno interessanti oppure già affrontate di recente su Commodore Computer Club .

Ci capita infatti, purtroppo, di esaminare schede in cui vengono richiesti argomenti già trattati sullo stesso fascicolo da cui la scheda è tratta!

Invitiamo pertanto i lettori, specie se principianti, a sfogliare con attenzione le pagine della nostra rivista e a leggere articoli che, pur se apparentemente non li interessano, contengono purtuttavia preziose informazioni su istruzioni particolari (POKE, PEEK, SYS ed altre) o tecniche insolite di programmazione, registrazione di dati, programmi eccetera.

Festeggiamo il 25mo numero di

commodore
COMPUTER CLUB
23 L. 3.000
La rivista degli utenti di sistemi Commodore

Molte riviste festeggiano la propria presenza in edicola ricordando ai lettori, periodicamente, la strada percorsa.

I numeri "chiave" coincidono, in genere, col 25mo anno (...nozze d'argento), 50 (d'oro) e pochi periodici, tra cui soprattutto quotidiani, vantano un'età ultracentenaria.

Commodore Computer Club, giunto al N.25, e approfittando del quarto anno di pubblicazioni, propone ai lettori affezionati (e nuovi) le copertine di alcuni numeri della rivista.

Il periodo di tempo trascorso vi sembra troppo breve per meritare una particolare attenzione? Forse, per una rivista, 25 numeri (e non anni) sembrano pochi per parlare di "storia"?.

Noi pensiamo che, nel campo dell'informatica, quattro anni di pubblicazione specializzata siamo paragonabili, per la Terra, al periodo di tempo di un'era geologica. Per convincervene, leggete le didascalie delle immagini disseminate in questo fascicolo...

Alessandro de Simone

Enduro MDX. La belle

zza elevata a potenza.

A partire da questo mese sarà data precedenza, per la pubblicazione, ai lettori che avranno inviato in unico "blocco" almeno dieci mini-programmi da una riga.

Ciò allo scopo evidente di semplificare il lavoro di scelta, già notevole, da parte della redazione.

I numerosi (ben 28!) microlistati di questo numero sono opera di Maurizio Dell'Abate. Se non specificato diversamente, possono esser digitati su qualsiasi computer Commodore. E' ovvio, comunque, che la parte del leone spetta allo starordinario Commodore 64.

#1

Interrupt 64 - Aumentando la frequenza di chiamata dell'interrupt IRQ e settando il flag di ripetizione tasti si ottiene questo simpatico effetto...

```
1 POKE 56325,1:POKE 65
  0,255
```

#2

Strisce 64 - Sul video compariranno delle strisce in movimento di caratteri maiuscoli e minuscoli, alternativamente; provando a tenere premuto un qualsiasi tasto il movimento cambia verso.

```
1 FOR I=1024 TO 2023:P
  OKE I,1:NEXT:FOR I=1
   TO 10000:POKE 53272
  ,21:POKE 53272,23:NE
  XT
```

#3

Fischio 64 - Facendo girare questa riga il SID emetterà un fischio acutissimo e straziante che si udirà fino a 200 metri di distanza se tenete il volume al massimo.

```
1 S=54272:FOR I=0 TO 2
  4:POKE S+I,0:NEXT:PO
  KE S+24,15:POKE S+6,
  240:POKE S,255:POKE
  S+1,255:POKE S+4,17
```

#4

Randomaze 64 - Viene disegnato sul video un labirinto pseudocasuale. E' interessante notare come in questo caso la casualità dia luogo ad un ottimo risultato; è una routine preziosa per i vostri videogames, ma è altrettanto divertente provarla immediatamente, magari seguendo con un dito il percorso.

```
1 FOR I=1024 TO 2023:P
  OKE I,77.5+RND(1):NE
  XT:WAIT 198,1
```

#5

Motore 64 - Si udirà il rumore di un motore che aumenta gradualmente la sua velocità fino a...

```
1 FOR I=1 TO 100:POKE
  54296,15:POKE 54296,
  0:FOR T=1 TO 100-I:N
  EXT:NEXT
```

#6

Superbiro 64 - Al centro del video comparirà un simpatico uccellino che batte velocemente le ali.

```
1 PRINT"[CLEAR]":POKE
  1524,81:FOR I=1 TO 1
  000:POKE 1523,85:POK
  E 1525,73:POKE 1523,
  74:POKE 1525,75:NEXT
```

#7

Colpo d'occhio - Comparirà per una frazione di secondo un numero da 0 a 10. Il vostro compito sta nell'individuarlo e nel rispondere correttamente alla domanda successiva.

```
1 N=INT(RND(1)*11):PRI
  NT"[CLEAR]";N:FOR I=
  1 TO 50:NEXT:INPUT "
  [CLEAR]QUALE";Z:IF Z
  =N THEN PRINT"ESATTO
  !!"
```

#8

Terremoto 64 - Fate girare questa linea e tutte le informazioni presenti sul video verranno distrutte a causa di un catastrofico terr.. anzi, videomoto. Si esce tramite RUN/STOP & RESTORE.

```
1 FOR I=0 TO 255:POKE
  53270,I:NEXT
```

#9

Randomusic 64 - Il SID emette suoni casuali che danno origine ad una musica molto ricca di armonia.

```
1 S=54272:POKE S+24,15
  :X=INT(99*RND(1)):PO
  KE S+8,X:POKE S+13,2
  55:POKE S+11,33:GOTO
   1
```

#10

Freccia - Una velocissima freccia percorrerà per qualche secondo il vostro schermo.

```
1 PRINT"[CLEAR]";:FOR
  I=0 TO 1000:PRINT"->
  ";:FOR G=1 TO 10:NEX
  T:PRINT"[2 LEFT]  ";
  :NEXT
```

#11

Riflessi 64 - Dopo il RUN il computer aspetterà un po' di tempo (sempre differente) e poi colorerà lo schermo di bianco. Appena avvertite il cambiamento cromatico premete immediatamente un tasto; se il tempo di risposta è buono (sarà raro) lo schermo viene pulito, altrimenti comparirà normalmente la scritta READY.

```
1 A=TI:B=53281:POKE B,
  6:FOR I=1 TO 999*RND
  (1):NEXT:POKE B,1:PO
  KE 198,0:WAIT 198,1:
  IF (TI-A)<30 THEN PR
  INT"[CLEAR]"
```

#12

Metronomo 64 - Un aiuto ai musicisti ed ai compositori: il metronomo. E' necessario dare in input la frequenza (0 - velocissimo; 20 - lentissimo) e il computer scandirà il tempo senza errori.

```
1 INPUT "[CLEAR]FREQ.
  0<20";F:FOR Z=1 TO 9
  999:FOR I=1 TO F*100
  :NEXT:POKE 54296,15:
  POKE 54296,0:NEXT
```

#13

Reset 64 - Premendo simultaneamente RUN/STOP & RESTORE non otterrete la reinizializzazione di alcune variabili (come al solito), ma il reset totale del sistema; perderete quasi irrimediabilmente il programma in memoria. Le istruzioni in questa riga modificano il puntatore alla routine di NMI settandolo per "guardare" dalla locazione 64738 in poi.

```
1 POKE 792,226:POKE 79
  3,252
```

#14

Orologio - Immettete nella variabile TI$ l'ora (formato: TI$T"HHMMSS"), cancellate lo schermo e date il RUN; avrete costantemente sul video l'ora esatta.

```
1 PRINT"[HOME]   [2 DO
  WN]";LEFT$(TI$,2);":
  ";MID$(TI$,3,2);":";
  RIGHT$(TI$,2):GOTO 1
```

#15

Geometria - Date in input la lunghezza della circonferenza e otterrete raggio, diametro e area del cerchio in questione.

```
1 INPUT "[CLEAR]CIRC";
  C:R=C/(2*π):D=R*2:A=
  R*R*π:PRINT"RAGG";R:
  PRINT"DIAM";D:PRINT"
  AREA";A
```

#16

Geometria 2 - Date in input la lunghezza del raggio ed otterrete circonferenza, diametro ed area del cerchio.

```
1 INPUT "[CLEAR]RAGG";
  R:C=R*2*π:D=R*2:A=R*
  R*π:PRINT"CIRC";C:PR
  INT"DIAM";D:PRINT"AR
  EA";A
```

#17

Geometria 3 - Date in input la superficie ed otterrete raggio, diametro e circonferenza del cerchio.

```
1 INPUT "[CLEAR]AREA";
  A:R=SQR(A/π):D=R*2:C
  =D*π:PRINT"CIRC";C:P
  RINT"DIAM";D:PRINT"R
  AGG";R
```

#18

Tabellina - Date in input il numero di cui volete ottenere la tabellina (es.: 3=3, 6, 9,...).

```
1 INPUT "[CLEAR][DOWN]
  [BIANCO]NUMERO";N:FO
  R I=1 TO 10:PRINTN*I
  :NEXT
```

#19

Voce - Incredibile!! un qualsiasi Commodore può parlare e pronunciare chiaramente la frase "BERE FA BENE" con questo programma. Da provare subito.

```
1 A$="BERE":B$="FA BEN
  E":A=5467:POKE A,96:
  SYSA:PRINTLEFT$(A$,2
  );"F";LEFT$(B$,2);"!
  "
```

#20

Text editor - Lungi dall'essere considerato un word processor, questo è un programma di scrittura. Il testo viene scritto sullo schermo e viene stampato a buffer pieno (ogni 80 caratteri) od alla prima pressione del tasto RETURN; purtroppo non è possibile fare correzioni (!). Premendo cursor down si otterrà la stampa in minuscolo e si tornerà in maiuscolo conn cursor up. Questa riga è l'ideale per scrivere testi brevissimi (titoli, ecc.) senza tirare in ballo un programma più impegnativo.

```
1 CLOSE 4:OPEN 4,4:GET
   A$:PRINTA$;:PRINT#4
  ,A$;:GOTO 1
```

#21

Leve - Viene dato in input il peso A, la sua distanza dal fulcro (braccio di A) ed il peso B. Si otterrà la lunghezza del braccio di B perchè la leva stia in equilibrio.

```
1 PRINT"[CLEAR][DOWN][
  RIGHT][NERO]A——F——
  B":INPUT "PESO A";P:
  INPUT "BR A";B:INPUT
   "PESO B";C:PRINT"BR
  B=";P*B/C
```

#22

Inverti parole - Questa riga inverte la parola o la frase data in input in tutti i sensi. Provare per vedere.

```
1 INPUT A$:FOR I=LEN(A
  $) TO 1 STEP -1:B$=B
  $+MID$(A$,I,1):NEXT:
  PRINTCHR$(18);"[DOWN
  ][BIANCO]";B$
```

#23

Nobreak 64 - Tramite questa routine si neutralizza il tasto RUN/STOP e quindi anche la combinazione RUN/STOP & RESTORE. E' possibile riabilitare il tutto con la POKE 808,237.

```
1 POKE 808,225:REM  DI
  SABILITA RUN/STOP
```

#24

Sinusoide - La riga genera una sinusoide in movimento composta da parole. Modificanndo la parola nel programma si può ottenere una titolatrice.

```
1 FOR A=0 TO 100 STEP
  .2:Y=SIN(A):PRINT TA
  B(16*Y+18);"TITOLO":
  NEXT
```

#25

Tricolore- Una coloratissima bandiera italiana viene disegnata a tutto schermo sul video del vostro Commodore.

```
  BIANCO]
    [ROSSO]
    ";:NEXT:WAIT 198,1
1 PRINT"[HOME]";:FOR I
  =0 TO 24:PRINT"[RVS]
  [VERDE]             [
```

#26

Rom 64 - La riga stampa sul video la scritta " **** COMMODORE 64 BAS..." presa direttamente dalla ROM.

```
1 FOR I=58483 TO 58520
  :PRINTCHR$(PEEK(I));
  :FOR T=1 TO 500:NEXT
  :NEXT
```

Maurizio Dell'Abate

Nota Bene

Alcune righe tra quelle pubblicate sembrano possedere più di 80 caratteri e, come tali, inaccettabili dal computer.

Nei casi in cui ci si accorga che la riga è troppo lunga, è necessario ricorrere alle abbreviazioni dei comandi così come indicato nell'appendice specifica riportata nel manuale del computer in vostro possesso.

Ad esempio invece di scrivere PRINT è possibile abbreviare col punto interrogativo (?). Invece di POKE potete scrivere il carattere "P" seguito dal carattere che viene visualizzato premendo contemporaneamente il tasto shift insieme con "O". Tutte le abbreviazioni possibili, lo ripetiamo, sono riportate in una delle appendici di qualsiasi manuale Commodore.

Nel caso sbagliate a digitare i microlistati che superano, in lunghezza, gli ottanta caratteri (SYNTAX ERROR o altri tipi di errore), è necessario, per sicurezza, ribatterli per intero e non apportare modifiche alla riga visualizzata con l'istruzione LIST.

ATTENZIONE!!!
A tutti i
Commodore Computer Club

Molti circoli si sono aperti e molti sono usciti allo "scoperto" dopo il nostro invito ad aprire un Computer Club, apparso sul N. 21 di C.C.C.

Allo scopo di rendere un servizio migliore ai nostri lettori che intendano contattare uno di questi simpatici circoli culturali, i segretari dei circoli stessi sono pregati di compilare il seguente tagliando, o sua fotocopia, e di inviarlo in busta chiusa (affrancata secondo le vigenti tariffe postali) a:

Systems Editoriale
Servizio Notizie Computer Club
Viale Famagosta, 75
Milano

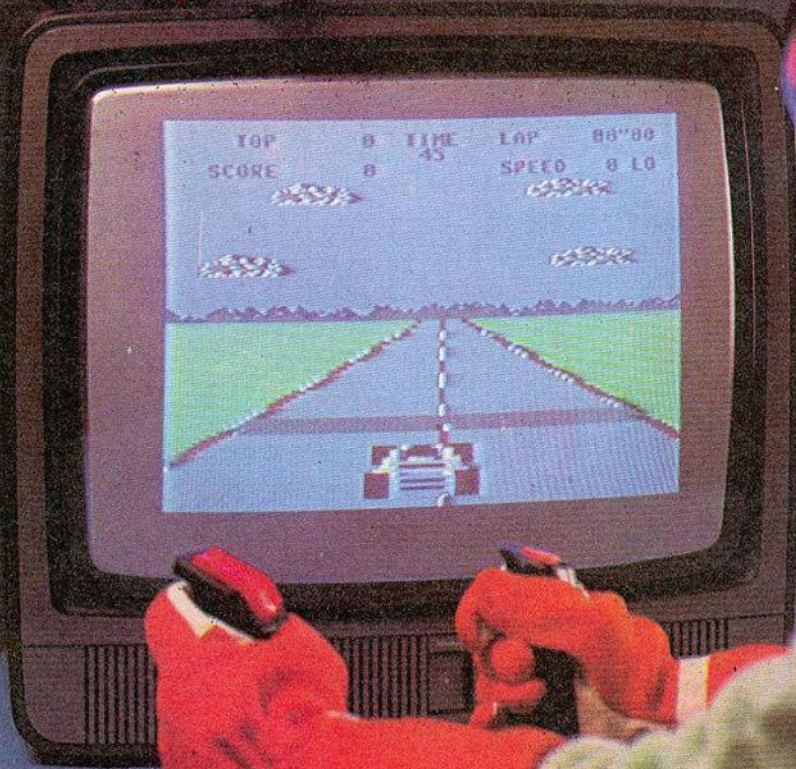
La completa compilazione dell'intera scheda, è INDISPENSABILE per la pubblicazione gratuita sulla nostra rivista.

Nome del club: .... Sede del club: .... Via ....
C.A.P. .... Città .... Prov.: .... Tel. Prefisso: ......N. ....
Presidente: .... Segretario: .... N. soci fondatori: ....
N. di soci finora iscritti: .... Data di fondazione: .... Giorni di apertura della sede: ....
Orario di apertura: .... Computers disponibili (specificare): ....
Periferiche disponibili (specificare): .... Programmi disponibili (N. approssimativo): ....
Videogiochi N.: .... Professionali N.: .... Biblioteca tecnica N. volumi: ....
N. abbonamenti a riviste italiane: .... N. abbonamenti a riviste straniere: .... Quota di iscrizione L. ....
(Specificare se annuale, mensile ecc.) ....
Attività previste: .... Bollettino periodico emesso: .... Attività già svolte: ....
Eventuali sponsor: .... Disponibilità alla sponsorizzazione (sì/no): ....

Il sottoscritto ...., presidente del Computer Club .... autorizzo la Systems Editoriale a diffondere notizie riguardanti le attività del circolo culturale citato anche se pervenute in redazione in via non ufficiale.
Dichiaro inoltre che le informazioni comunicate corrispondono al vero e che, in caso di difformità accertata da parte di incaricati della Systems stessa, Commodore Computer Club, allo scopo di tutelare la buona fede degli utenti della rivista, si riserva il diritto - dovere di avvertire i propri lettori nel modo e nella forma che riterrà più opportuni.

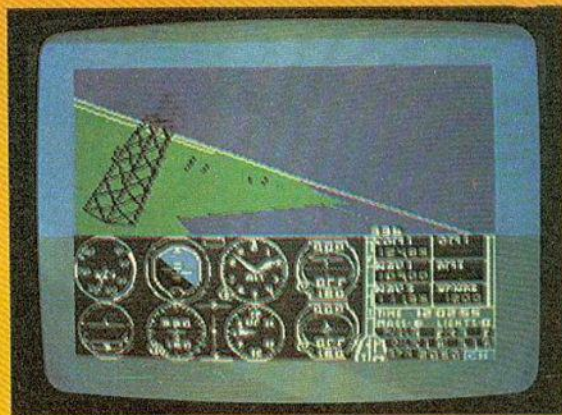
In fede

SPECIALE SIMULAZIONI

La simulazione può essere considerata una vera e propria scienza. Cosa succede se... Questa è la base di qualsiasi tipo di simulazione. L'ausilio del computer ha enormemente aiutato gli operatori di questo settore a partire da applicazioni piuttosto banali alla realizzazione di un linguaggio ideale per la simulazione bellica: l'ADA. La computer-grafica infine ha portato la simulazione a livelli eccelsi. E ... con il Commodore? Il Flight Simulator II ne è un validissimo esempio.

SIMULANDO SI IMPARA...
CON IL COMMODORE 64/128

Il Flight Simulator II sul Commodore 64 e sul C128 oltre a divertire può aiutare in una comprensione della problematica del volo effettuato su un piccolo aereo da turismo. Non pensate però che, divenuti maghi del joystick, ne possiate pilotare uno vero.

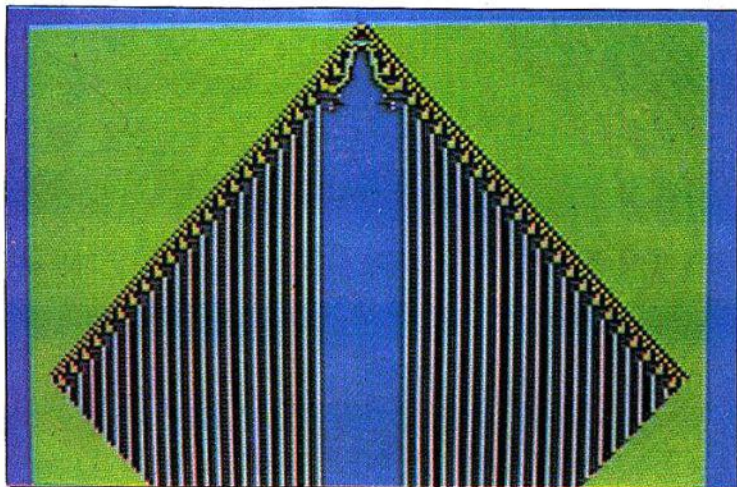
Con uno dei due joystick si può intervenire sull'assetto del vostro monomotore. Quindi si possono eseguire ottime virate, loop ecc.

Il primo approccio con il Flight Simulator II è proprio inerente alla fase di decollo. Siete sulla pista. Un pò di rullaggio, azione sui flaps, date gas e ... via un bel giro fra le nuvole.

Il Flight Simulator II offre anche la possibilità di pilotare un aereo da guerra contro un avversario temibile. Una simulazione che mette a dura prova sia la vostra bravura di piloti che di abili combattenti.

Italia - Evoluzione del tempo sulla nostra penisola nell'arco delle 24 ore mediante visualizzazione di zone diversamente colorate.

Simulazione dell'evoluzione delle condizioni meteriologiche sull'Europa, mediante visualizzazione di una perturbazione che si sposta della Spagna fino verso le nostre regioni.

Automa cellulare

Computer-grafica e simulazione: un binomio di sicuro successo e con enormi prospettive.

Il programma presentato appartiene alla specie degli "automi cellulari", organismi che stanno rapidamente invadendo i circuiti dei calcolatori e sono sempre più utilizzati nel campo della simulazione al computer.

Ve ne propongo uno: spero vi sorprenda piacevolmente e stimoli a studiare più a fondo il comportamento di questo strano "animale".

Come ogni organismo, esso ha origine da un'unica cellula madre (un pixel colorato sul video), la quale si evolverà dando origine ad una struttura piramidale più o meno variopinta.

Far girare il programma è molto semplice: caricate le routines grafiche di Danilo Toma (già pubblicate sul N. 14 di C.C.C.) e date il RUN. Subito dopo copiate il listato pubblicato su queste pagine e lanciatelo.

Vi verrà chiesto di immettere un codice, che potremmo impropriamente definire DNA, il quale può essere un numero a piacere (con alcune limitazioni che sono comunque chiarite all'interno del programma). Battete il tasto Return e attendete un tempo proporzionale al numero di righe impostato (secondo INPUT).

Pur utilizzando le ottime routines grafiche, la "creatura" andrà crescendo con una velocità che a qualcuno non sembrerà proprio eccessiva. Questo a causa dei numerosi calcoli che il programma deve svolgere per determinare il colore e la posizione di ogni punto e il tutto si traduce con una spesa di qualche minuto per ogni disegno.

Per uscire dalla pagina grafica, una volta che il disegno è terminato, premete un tasto.

L'adattamento al C-16

Pur essendo rivolto espressamente al C64, sono sicuro che i possessori di C16 e PLUS4 sapranno facilmente adattarlo.

I punti su cui intervenire sono le poche linee nelle quali operano i comandi per la gestione della pagina in alta risoluzione. Cambiate la sintassi dei comandi CLEAR, COLOR, TEXT, PLOT e l'origine delle coordinate che, per le routines grafiche, è leggermente diversa: correggete adeguatamente i valori di W e C del listato.

Sperimentazione e simulazione

E' questo un aspetto affascinante della programmazione. Oggi sempre più scienza e industria sposano la causa della simulazione, con indubbi benefici: ampiamento delle possibilità di ricerca e considerevoli risparmi di denaro.

Un esempio: nella fisica delle particelle sono da tempo conosciute le leggi che regolano il moto degli elettroni. Si può allestire facilmente un esperimento in laboratorio per studiarne il comportamento all'interno del campo magnetico presente in un cinescopio. Ma in nessun caso saremmo in grado di studiare in "vitro" cosa succede all'interno di una stella di neutroni: le condizioni sono così estreme che è chiaramente impossibile riprodurle artificialmente anzi, in situazioni così spinte anche le moderne teorie fisiche cominciano a vacillare.

Dunque ci si deve rassegnare ad applicare stoicamente le equazioni relativistiche ed estrapolare manualmente i risultati?

Fortunatamente no.

Naturalmente la cosa non è che ci riguardi da vicino ma, con l'ausilio di un piccolo computer e del software adeguato, anche noi incompetenti e squattrinati potremmo se non altro appagare la nostra curiosità e vagare all'interno di un acceleratore di particelle o immaginare di essere un ricco signorotto di una città medioevale. E ancora viaggiare per lo spazio-tempo, buchi neri, quasar...

Il fiocco di neve

Torniamo con i piedi per terra e vediamo cosa possiamo fare con il programmino proposto.

Questo è un noto esempio di simulazione grafica conosciuto sotto il nome di "automa cellulare". Per capire come funziona, provate ad immaginare come si forma un fiocco di neve. Sicuramente tutti ne conoscete la sagoma regolare e piacevole.

Non è altro che l'immagine ingrandita al microscopio di un cristallo di ghiaccio e riportato spesso ad esempio di come fenomeni che appaiono casuali e disordinati abbiano invece un intimo e rigoroso ordine.

La simulazione al computer dello sviluppo di un fiocco di neve è abbastanza semplice: nel modello il piano è suddiviso in un reticolo di piccole cellule esagonali; a ciascuna viene assegnato il valore 0, che corrisponde al vapore acqueo, oppure 1 che corrisponde al ghiaccio. Partendo da un'unica cellula generatrice, il cristallo si accresce ed il valore di ogni altra viene calcolato in funzione dello stato delle compagne vicine.

Convertendo le informazioni numeriche in immagini, assegnando ad esempio al valore 0 il colore nero ed al valore 1 il blu, il risultato è veramente sorprendente.

In natura fenomeni simili sono numerosissimi; non solo a livello microscopico e inanimato, ma estendendosi senza eccezioni al mondo macroscopico e del vivente.

Basti pensare alla stupenda colorazione di molte conchiglie o alla pigmentazione della pelle di alcuni animali.

Il nostro piccolo automa simula un processo di accrescimento molto simile a quello esistente in natura, nel quale ogni singola cellula è condizionata dalle sue immediate vicine.

Ad ogni punto è assegnato un valore numerico dipendente dal proprio colore:

0 = nero
1 = rosso
2 = blu
3 = giallo

Non è obbligatorio usare proprio questi colori e, volendo, è possibile cambiarli.

Il colore viene calcolato facendo la somma delle tre cellule poste al di sopra del punto in esame, seguendo un determinato codice (CD):

543210 somme
CD = 312010 colori

Partendo da sinistra verso destra: se la somma vale 5 allora il colore sarà il 3 (giallo). Se vale 4 il colore sarà l'1 (il rosso) e così via...

N.B. I disegni più graziosi si ottengono assegnando un numero pari di caratteri 0, 1, 2 oppure 3.

Esempio: 00122100 (=8 caratteri).

Flavio Molinari

```
100 REM  AUTOMA CELLULARE
110 REM  PER COMMODORE 64
150 :
160 REM  BY FLAVIO MOLINARI
170 :
180 REM  CARICARE PRIMA LE
190 REM  RUOTINES GRAFICHE
200 REM  I DANILO TOMA
280 :
290 ←TEXT 6,14:REM  MODO TESTO COL
    ORI ORIGINALI
300 PRINT CHR$(147):REM  CLEAR
310 PRINT"AUTOMA CELLULARE ";CD$:P
    RINT
320 PRINT"DEVI BATTERE UN CODICE C
    OMPOSTO DA UN"
330 PRINT"MAX DI 10 CIFRE."
340 PRINT"OGNI CIFRA PUO' AVERE UN
    O DI QUESTI"
350 PRINT"VALORI: 0,1,2,3":PRINT:P
    RINT"PROVA, AD ESEMPIO:"
351 PRINT"3211310"
352 PRINT"3130220"
353 PRINT"3311100320"
354 PRINT"2213310"
355 PRINT"1001"
360 PRINT:CLR :INPUT "CODICE";CD$
365 INPUT "N.RIGHE (15-150)";NR
370 LR=LEN(CD$)
380 IF  LR=0  OR  LR>10  THEN  600
390 FOR  Q=0  TO  LR-1
400 R(Q)=VAL(MID$(CD$,LR-Q,1))
410 IF  R(Q)>3  THEN  590
420 NEXT
430 :
440 W=160:REM  ASSE X
450 C=100:REM  ASSE Y
460 ←CLEAR:REM  PULISCE PAGINA GRA
    FICA
470 ←MGRAF 0,2,6,7:REM  COLORI
480 DIM  A(320)
490 A(W)=1:REM  PUNTO INIZIALE
500 :
510 FOR  Y=1  TO  NR:B=0
520 GET  Q$:IF  Q$=CHR$(13)  THEN
     290
530 FOR  Q=W-Y  TO  W+Y
540 A=R(A(Q-1)+A(Q)+A(Q+1))
550 A(Q-1)=B:B=A
560 ←COL OR  A:←PLOT Q-W,C-Y,0
570 NEXT:A(Q-1)=A:NEXT:PRINTCHR$(1
    47)
574 ←TEXT6,14:PRINT"PER CONTINUARE
     PREMI UN TASTO"
576 GET A$:IF A$="" THEN 576
580 ←MGRAF 0,2,6,7
581 FOR  Y=NR-1  TO 1  STEP  -1:B=
    0
583 FOR  Q=W-Y  TO  W+Y
584 A=R(A(Q+1)+A(Q)+A(Q-1))
585 A(Q-1)=B:B=A
586 ←COL OR  A:←PLOT Q-W,C+Y-NR*2,
    0
587 NEXT:A(Q-1)=A:NEXT
590 GET A$:IF A$="" THEN 590
595  RUN
600 PRINT"CODICE ERRATO":END
```

Gestione di un mazzo di carte

Come simulare, su un qualsiasi computer, la creazione, il mescolamento, la distribuzione e la visualizzazione di un mazzo di 40, 52 o qualsiasi altro numero di carte.

Sui computers si trovano quasi sempre giochi stressanti e così impegnativi che piuttosto che procurare distensione causano esaurimenti nervosi.

E pensare che solo fino a una quindicina di anni fa il divertimento più diffuso nelle serate, sopratutto quelle uggiose, era costituito da una rilassante partita a carte.

Questo gioco sembra oggi riservato ai simpaticissimi nonnetti che, nella maggior parte dei casi, dichiarano di non capir niente di queste cose strane (computer) che procurano solo rompicapi.

Per chi avesse fantasia e nostalgia di una serata di vero divertimento, proponiamo un breve listato che contiene varie soubroutines in grado di gestire un mazzo di carte, per creare tutti i vostri giochi, così magari potrete accostare qualche arzillo vecchietto al favoloso mondo dei computers.

come si usano le subroutines

Digitate dapprima il listato n. 1 (numerato da 10000 in poi, tanto per intenderci).

Per iniziare a gestire un mazzo di carte è ovviamente necessario crearlo eseguendo una subroutine che dovrete sempre "chiamare" per prima nei programmi in cui vorrete applicare la tecnica descritta in questo articolo.

La routine in oggetto è: GOSUB 63560

Prima di chiamarla, potete modificare alcune variabili per controllare la struttura delle carte.

Se nella variabile CT ponete il valore 10 (=numero di carte per ciascuno dei quattro semi), sarà creato un mazzo da 40 carte. Con CT=13 lo stesso sarà da 52 carte.

Modificando la variabile CO$, che contiene esclusivamente il carattere-colore, potrete segliere il colore del bordo delle carte; il colore standard scelto è il blu chiaro.

Infine, cambiando i colori nella riga 63900 cambierete i colori dei semi delle carte.

Nella versione pubblicata il colore dei semi delle carte sono il bianco e il nero invece che rosso e nero, ciò è dovuto al fatto che il rosso, su molti televisori e monitors, è illeggibile su sfondo blu.

Se vi serve un mazzo di carte mischiate potete crearlo con:
GOSUB63270

Il mazzo mischiato verrà posto nel vettore MA(X).

La formula per ottenere una carta dal valore contenuto nel vettore è:

```
SE=INT(MA/CT)
VA=MA-SE*CT
SE=SE+1
VA=VA+1
```

in cui MA è il valore contenuto nel vettore, SE il seme della carta e VA il valore della stessa.

Per chi non avesse capito la formula niente di grave: esiste una subroutine che estrae automaticamente le carte dal mazzo una di seguito all'altra; per utilizzare questa possibilità dovete dare:
GOSUB 63450:PRINT SE;VA

così i valori verranno aggiornati automaticamente.

Infine per stampare le carte fate:
GOSUB 63080
valori si controllano con:

Y=posizione verticale della carta.
X=posizione orizzontale della carta.
SE=seme della carta (0=carta coperta).
VA=valore della carta.

Si noti che la carta 10 è indicata con I0; ciò viene utilizzato per evitare di confondere la carta 1 con la carta 10 nelle sovrapposizioni.

Per sovrapporre parzialmente le carte sarà opportuno stamparle da sinistra verso destra e dall'alto verso il basso in modo da lasciare scoperta la parte in alto a sinistra della carta, che ne permette il completo riconoscimento (valore e seme).

Le righe comprese tra 10000 e 15010 servono solo per controllare il buon funzionamento delle subroutines e non sono di alcuna utilità per il loro funzionamento; quindi in caso di utilizzo delle subroutines sarebbe opportuno cancellarle.
Inoltre è pubblicato un simpaticissimo programma (listato DEMO) di esempio per l'uso delle carte. E' necessario digitarlo DOPO aver caricato nel computer (o digitato) il programma contenente le subroutine di gestione (da 10000 in poi).

Studiandolo attentamente potrete meglio comprendere il funzionamento delle subroutines che si prestano a facili modifiche.

N.B. Se, dando il RUN, non dovesse apparire nulla, è probabile che il colore del fondo e delle carte (semi, bordi, valori) siano eguali. In questi casi, prima di dare il RUN, cambiate il colore del cursore (tasto CTRL insieme con uno dei tasti da 1 a 8).

Trovata, in tal modo, la combinazione soddisfacente, inserite nelle prime righe di programma il comando idoneo per il cambio del colore del cursore.

Buon divertimento. **Fabio Sorgato**

```
10000 REM  GESTIONE DI UN
10001 REM  MAZZO DI CARTE
10002 :
10003 REM  DI FABIO SORGATO
10004 :
10007 PRINTCHR$(147)"ATTENDERE":CO$=
      CHR$(154):CT=10
10008 GOSUB 63560:PRINTCHR$(147);
10009 PRINT"MAZZO DA 40 CARTE"
10010 FOR SE=0 TO 4
10020 FOR VA=1 TO CT
10030 Y=(SE+1)*4
10040 X=(VA-1)*2
10050 GOSUB 63080
10100 NEXT:NEXT
10500 GOSUB 15000
11000 PRINTCHR$(147)"ATTENDERE"
11005 GOSUB 63270:PRINTCHR$(147);
11007 PRINT"MAZZO DA 40 CARTE MISCHI
      ATE"
11010 FOR K1=1 TO 4
11020 FOR K2=1 TO CT:GOSUB 63450
11030 Y=(K1+1)*4
11040 X=(K2-1)*2
11050 GOSUB 63080
11100 NEXT:NEXT:GOSUB 15000:CLR
12000 PRINTCHR$(147)"ATTENDERE":CO$=
      CHR$(154):CT=13
12005 GOSUB 63560:PRINTCHR$(147);
12007 PRINT"MAZZO DA 52 CARTE"
12010 FOR SE=0 TO 4
12020 FOR VA=1 TO CT
12030 Y=(SE+1)*4
12040 X=(VA-1)*2
12050 GOSUB 63080
12100 NEXT:NEXT
12500 GOSUB 15000
13000 PRINTCHR$(147)"ATTENDERE"
13005 GOSUB 63270:PRINTCHR$(147);
13007 PRINT"MAZZO DA 52 CARTE MISCHI
      ATE"
13010 FOR K1=1 TO 4
13020 FOR K2=1 TO CT:GOSUB 63450
13030 Y=(K1+1)*4
13040 X=(K2-1)*2
13050 GOSUB 63080
13100 NEXT:NEXT:GOSUB 15000:END
15000 PRINT:PRINT"PREMI UN TASTO"
15005 GET A$:IF A$="" THEN 15005
15010 RETURN
63000 REM *********************
63010 REM *SUBROUTNE CHE STAMPA*
63020 REM * LE CARTE. RICORDA: *
63030 REM * X,Y=COORINATE      *
63040 REM * SE =SEME CARTA     *
63050 REM *     (0=COPERTA)    *
63060 REM * VA =VALORE CARTA   *
63070 REM *********************
63080 FOR T=1 TO 5
63090 PRINTCHR$(19);LEFT$(YY$,Y-1+T)
      ;LEFT$(XX$,X);
63100 PRINTCA$(SE,VA,T);:NEXT:RETURN
63200 REM *******************
63210 REM *  SUBROUTINE CHE  *
63220 REM * MISCHIA LE CARTE *
63230 REM * E PONE  IL MAZZO *
63240 REM *  OTTENUTO NELLA  *
63250 REM *  MATRICE  MA(X)  *
63260 REM *******************
63270 FOR CA=1 TO CT*4:MI(CA)=0:MA(C
      A)=0:NEXT
63280 FOR CA=1 TO 4*CT
63290 RR=INT(RND(1)*CT*4)
63300 IF MI(RR)=0 THEN MI(RR)=1:MA(C
      A)=RR:NEXT:CO=0:RETURN
63310 GOTO 63290
63400 REM **********************
63410 REM *  ESTRAE  UNA CARTA  *
63420 REM *       DAL MAZZO     *
63430 REM **********************
63450 CO=CO+1:IF CO>CT*4 THEN RETURN
63460 MA=MA(CO)
63470 SE=INT(MA/CT)
63480 VA=MA-SE*CT
63490 SE=SE+1:VA=VA+1:RETURN
63500 REM **********************
63510 REM *   CREA  LE  CARTE   *
63520 REM *  DEI  VARI SIMBOLI  *
63530 REM *CON I VALORI COMPRESI*
```

```
63540 REM * TRA L'ASSO  E IL RE *
63545 REM * CT=10:40 CARTE      *
63546 REM * CT=13:52 CARTE      *
63550 REM ***********************
63560 IF CK=1 THEN RETURN
63565 RESTORE :CK=1:IF CT<>10 THEN C
      T=13
63570 READ A$:IF A$<>"***" THEN 6357
      0
63580 DIM VA$(13),CA$(4,13,5),MI(13*
      4),MA(13*4)
63590 FOR T=1 TO 4:READ SI$(T):NEXT
63595 IF CT=10 THEN FOR T=1 TO 13:RE
      AD A$:NEXT
63600 FOR T=1 TO CT:READ VA$(T):NEXT
63610 BO$=CHR$(194)
63620 SG$=CHR$(192)
63630 IF CO$="" THEN CO$=CHR$(154)
63640 CP$=CHR$(166)
63650 FOR SI=1 TO 4
63660 SI$=SI$(SI)
63670 FOR VA=1 TO CT
63680 VA$=VA$(VA)
63690 CA$(SI,VA,1)=CHR$(213)+SG$+SG$
      +SG$+CHR$(201)
63700 CA$(SI,VA,2)=BO$+VA$+CHR$(-32*
      (LEN(VA$)=1))+SI$+CO$+BO$
63710 CA$(SI,VA,3)=BO$+CHR$(32)+SI$+
      CO$+CHR$(32)+BO$
63720 CA$(SI,VA,4)=BO$+SI$+CO$+CHR$(
      -32*(LEN(VA$)=1))+VA$+BO$
63730 CA$(SI,VA,5)=CHR$(202)+SG$+SG$
      +SG$+CHR$(203)
63740 NEXT
63750 NEXT
69760 REM ***********************
63770 REM *CREA LE CARTE COPERTE*
63780 REM *    CHE  AVRANNO IL  *
63790 REM * SEGNO UGUALE A ZERO *
63800 REM *E UN VALORE  COMPRESO*
63810 REM *    TRA  UNO E CT    *
63820 REM ***********************
63830 FOR VA=1 TO CT
63840 CA$(0,VA,1)=CO$+" ╭"+SG$+SG$+SG
      $+"╮"
63850 CA$(0,VA,2)=BO$+CP$+CP$+CP$+BO
      $
63860 CA$(0,VA,3)=CA$(0,VA,2)
63870 CA$(0,VA,4)=CA$(0,VA,2)
63880 CA$(0,VA,5)=" ╰"+SG$+SG$+SG$+"╯
      "
63890 NEXT:PRINTCO$;
63900 DATA ***,"[NERO]♠","[CELESTE]♥
      ","[CELESTE]♦","[NERO]♣":REM
      NERO(TASTO CTRL.+1), CELESTE(C
      OMM.+7)
63910 DATA 1,2,3,4,5,6,7,8,9,10,J,Q,
      K
63920 DATA 1,2,3,4,5,6,7,J,Q,K
63950 FOR JK=0 TO 39:YY$=YY$+CHR$(17
      ):XX$=XX$+CHR$(29):NEXT:RETURN
```

Il N.1

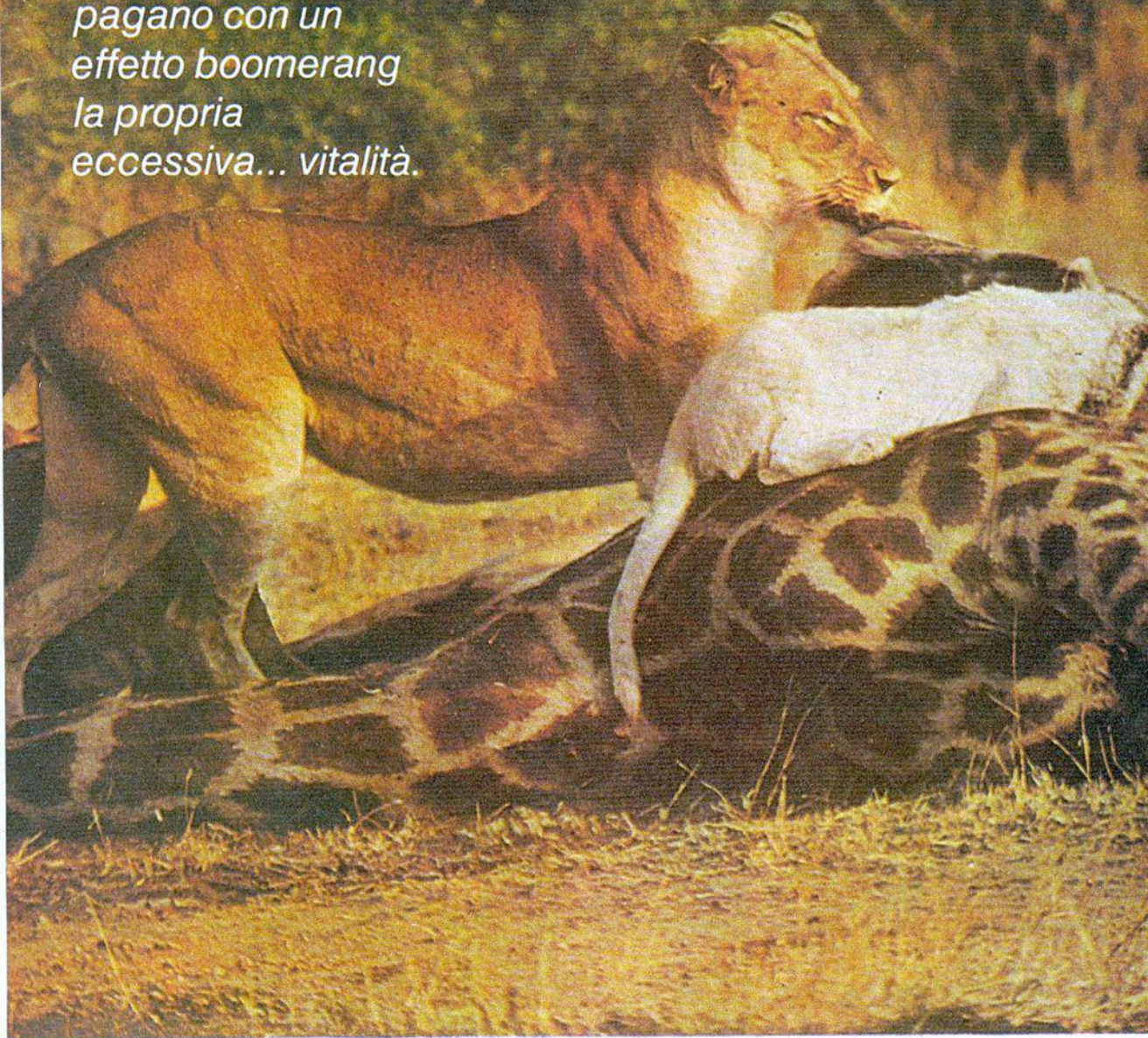
In occasione dello Smau '82, e dell'insediamento ufficiale in Italia della Commodore, viene pubblicato il primo numero di Commodore Computer Club, che rappresenta, nel nostro Paese, la prima rivista di informatica "verticale", dedicata, cioè, esclusivamente a prodotti Commodore. La coraggiosa iniziativa viene vista con scetticismo negli ambienti specializzati. "E' troppo presto - dicevano - per proporre al pubblico italiano una rivista specifica per prodotti di una sola marca".

I fatti, come è noto, hanno smentito senza appello queste affermazioni troppo affrettate.

In questo numero viene presentata, in anteprima (!), la scheda tecnica del Commodore 64. Questo computer, infatti, è pressochè sconosciuto in Italia e i programmi pubblicati possono girare, infatti, solo sul Vic 20 e sui modelli professionali Commodore PET.

La prima dispensa del mini-corso per imparare a programmare col Vic, alcuni videogiochi in Basic, la presentazione di un paio di package professionali, costituiscono il nocciolo del primo numero, oltre alla introduzione al codice macchina.

Predatori & prede

Semplice simulazione di un semplice ecosistema in cui i predatori, troppo voraci (e prolifici), pagano con un effetto boomerang la propria eccessiva... vitalità.

SIMULAZIONI

QUALSIASI COMPUTER

Esistono molteplici modelli matematici che descrivono il delicato equilibrio dinamico esistente tra predatori e prede che occupano, contemporaneamente, lo stesso territorio.

La complessità di tali modelli è direttamente proporzionale al grado di raffinatezza desiderato. Uno dei piu famosi è il modello di Lotke-Volterra, le cui soluzioni delle equazioni, costituenti il modello, mostrano una variazione ciclica del numero di individui, siano essi predatori oppure prede. Nella figura è riportata una di tali soluzioni.

Abbondanza di prede equivale ad abbondanza di cibo per i predatori, che in questo modo aumentano di numero. Ma molti predatori mangiano molte prede, quindi diminuiscono queste ultime, il che vuol dire scarsità di cibo e morte di molti predatori e così via.

Ogni volta che ci si trova di fronte ad un modello matematico, è lecito chiedersi fino a che punto funziona; quanto, cioè, siano affidabili le sue previsioni e se si renda necessaria una verifica che può richiedere anni. Ecco che viene in aiuto il computer, per mezzo del quale possiamo simulare un semplice ecosistema. Vediamo sotto quali condizioni l'evoluzione segue l'andamento ciclico previsto dalle equazioni di Lotke-Volterra.

Considerazioni preliminari

Bisogna notare che l'andamento ciclico è il risultato di determinate condizioni iniziali. Se, ad esempio, mettiamo 100 squali e 10 pesciolini in una piscina, ci aspettiamo che gli squali mangino subito i pesci, rimanendo, di conseguenza, senza più cibo. Quindi questa situazione "limite" evolve verso la completa estinzione di entrambe le specie. Anche predatori troppo prolifici possono portare alla stessa conclusione. E' chiaro quindi, che l'andamento ciclico può ottenersi solo con un determinato equilibrio tra molteplici fattori.

Per realizzare un buon programma di simulazione è necessario un compromesso tra raffinatezza di calcolo e velocità di elaborazione e non è facile trovare il giusto equilibrio. Per questo motivo il programma proposto prevede l'impostazione da input di un parametro che permette di variare la velocità di elaborazione a discapito dell'accuratezza.

Il territorio su cui le prede ed i predatori debbono muoversi è ovviamente lo schermo del computer, le cui dimensioni possono essere cambiate. Più lo schermo (territorio) è piccolo, più veloce è il programma (minor numero di "animali" da controllare) ma più approssimativa risulta la simulazione e viceversa.

La rappresentazione semigrafica

Scegliamo allora di rappresentare i predatori con una croce [CHR$(43)], e le prede con un puntino [CHR$(46)]. Questa rappresentazione serve, a noi osservatori, per seguire lo svolgimento dell'azione.

Il calcolatore, invece, si trova più a suo agio con i numeri. La matrice P(I,J) è l'immagine numerica dello schermo. Ogni elemento della matrice rappresenta un punto del territorio che può essere occupato da una preda o da un predatore.

Le ipotesi della simulazione

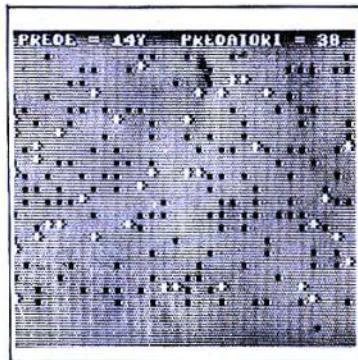
Dobbiamo a questo punto definire delle regole per il movimento, per la riproduzione, per la nutrizione e per la morte dei personaggi che ci accingiamo a collocare sullo scenario appena preparato. Le possibilità sono molte, ma forse è meglio non fare le cose troppo complicate.

Supponiamo che gli unici movimenti possibili siano quelli verso NORD, SUD, EST ed OVEST. Le prede si possono spostare solo verso posizioni vuote. I predatori invece anche verso posizioni occupate da prede, nel qual caso si nutrono del malcapitato. Non possiamo presupporre una riproduzione di tipo sessuale, perchè definire i due sessi e stabilire le regole di corteggiamento renderebbe il programma troppo complesso (e lento).E' comunque un'idea per una possibile sofisticazione. Allora accontentiamoci di specie partenogeniche.

Dobbiamo definire due periodi di riproduzione, uno per le prede, che generalmente sono più prolifiche, ed uno per i predatori; dopo un tempo prefissato si ha la generazione di un nuovo individuo. Inoltre bisogna anche definire un tempo di inedia per i predatori, cioè il tempo massimo di sopravvivenza senza cibo. Questi saranno i parametri di ingresso della simulazione.

Il tempo scorre con salti discreti, che chiameremo crononi.

I predatori

I predatori, quando nascono, sono rappresentati ciascuno, nella matrice P(I,J), dal numero 1000. Questo viene

incrementato di 1 ad ogni cronone e quando raggiunge il valore di riproduzione viene generato un nuovo individuo. Il neonato va ad occupare una posizione, se libera da altri predatori, adiacente al genitore. Se tutte le posizioni sono occupate l'individuo non si muove e non si riproduce. Dopo questa operazione, sia il genitore che il figlio sono posti a 1000. Oltre all'età ed alla posizione, è necessario, per quanto riguarda i predatori, tenere in conto del tempo di inedia. Questo viene fatto mediante la matrice IN(I,J), che, alla posizione (I,J), contiene 0 se in corrispondenza in P(I,J) vi è una preda o vi è nulla, altrimenti vi è il numero di crononi passati dall'ultimo pasto. Quindi ogni volta che si sposta un predatore bisogna spostare ed aggiornare anche il contenuto di IN(I,J). Se trova da mangiare viene posto a zero, altrimenti va incrementato di 1. Se viene raggiunto il massimo, vuol dire che il predatore è morto di fame, quindi deve essere eliminato dallo schermo.

Le prede

Per le prede le cose sono più semplici. Infatti non abbiamo problemi di inedia perchè, per semplicità, supponiamo che il territorio in esame (o pianeta) abbia abbondanza di cibo per le prede.

Le prede appena nate le rappresentiamo con il numero 10, che viene incrementato e controllato, ad ogni cronone, in maniera perfettamente analoga ai predatori. Raggiunta l'età della riproduzione, nasce un nuovo individuo. Sia il nuovo nato che il genitore vengono posti a 10.

Inizialmente il programma deve popolare il pianeta, sono allora richiesti i parametri iniziali riguardanti la densità di popolazione e la percentuale di predatori rispetto alle prede.

Il programma

Viene preparata la matrice P(I,J) secondo le precedenti indicazioni (linee 280-320). Si noti che anche l'età viene inizialmente attribuita casualmente, perchè, altrimenti, si avrebbe un innaturale raddoppio ciclico della popolazione. Le linee 335-380 visualizzano la situazione. Si scorre la P(I,J) e si accende il simbolo di preda, quando il contenuto è compreso tra 10 e 1000, oppure il simbolo di predatore, quando è superiore a 1000, altrimenti si lascia lo spazio vuoto.

Dopo questa fase entriamo nel cuore della simulazione (linee 390-500). Bisogna passare in rassegna P(I,J) per trovare le locazioni occupate da un qualsiasi individuo. Una volta trovate è necessario cercare, nei relativi dintorni, i posti occupati, che ovviamente non possono essere raggiunti. Ricordiamo che, dal punto di vista dei predatori, una locazione è non raggiungibile solo se occupata da un altro predatore. La variabile "C" serve allo scopo. Infatti assume un valore differente per ogni configurazione di posti occupati. Ad esempio, se NORD e SUD non sono liberi, cioè P(I+1,J) e P(I-1,J), allora "C" assume valore 3; di conseguenza alla linea 450 verrà eseguito il terzo sottoprogramma, che sposta casualmente l'individuo in esame (se è una preda) in una delle posizioni libere, in questo caso EST o OVEST. Ricordiamoci che gli spostamenti possibili sono 4 (N,S,E,O), quindi necessitiamo di 4 routine che li implementano, e sono precisamente quelle che iniziano alle linee 590 605 620 635. Inoltre per ogni configurazione di posti liberi è necessario scegliere casualmente tra una, due, tre o tutte e quattro queste routine, L'operazione è svolta dalle linee 510-585. Le istruzioni da 655 alla fine svolgono la stessa funzione per i predatori. Questi devono essere trattati in maniera diversa, in quanto possono raggiungere anche posizioni occupate da prede. La variabile "S" permette di distinguere predatori da prede. Il modo di operare è analogo a quello di "C". La matrice M(I,J) serve per evitare che un individuo venga spostato due volte durante lo stesso cronone. Se infatti M(I,J)=1, vuol dire che l'individuo è stato spostato durante il cronone corrente. Le variabili XX e YY hanno funzione di contatori delle prede e dei predatori. Il loro valore è continuamente visualizzato sulla prima riga dello schermo.

Suggerimenti per sofisticazioni

Una buona idea sarebbe quella di memorizzare il numero di predatori e prede per ogni simulazione (tschermata), utilizzando il registratore oppure il disk drive, e, dopo un centinaio di crononi, "graficarli" con istogrammi ed effettuare un confronto con gli andamenti previsti dalle equazioni Lotke-Volterra.

Ci auguriamo che quanto detto risulti chiaro, in modo che coloro che sono interessati possano ampliare il programma con le modifiche che ritengono più opportune.

Per finire un dettaglio per la digitazione del programma. Gli unici simboli grafici presenti si ottengono con i tasti del cursore CRSR-LEFT, CRSR-DOWN e CLR-HOME eventualmente premuti insieme a SHIFT.

Come gira il programma

Come è stato detto, il programma pubblicato gira su qualsiasi computer a patto di effettuare le poche modifiche relative alle variabili indicanti la prima locazione di memoria schermo, la colorazione del bordo e del fondo del video.

Allo scopo di velocizzare la simulazione stessa (accontentandosi, ovviamente, di una minore accuratezza della simulazione), provate a digitare il valore 7 alla richiesta del numero delle righe e delle colonne. Ogni simulazione, con valori così bassi, richiede pochissimi secondi.

Volendo, invece, una maggiore accuratezza, si rende necessario ricorrere alle dimensioni massime dello schermo. In questo caso si consiglia di compilare il programma in modo da ridurre drasticamente il tempo necessario per ciascuna simulazione.

Programma in Pascal

Gli interessati, che posseggono il linguaggio Oxford Pascal per il C-64, possono consultare il N.12 della rivista Commodore (sempre della Systems editoriale) su cui è pubblicata una simulazione analoga dal nome WA-TOR.

Antonio Visconti

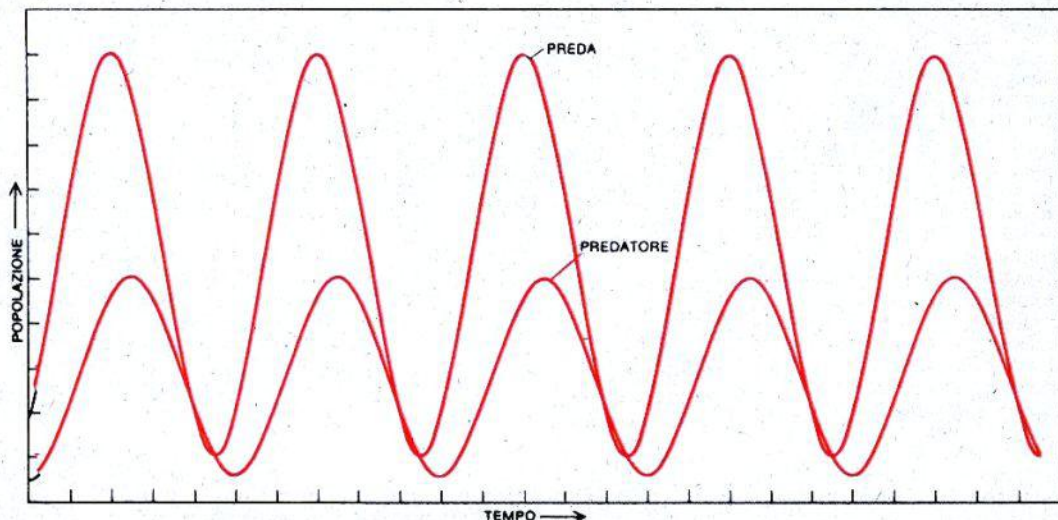



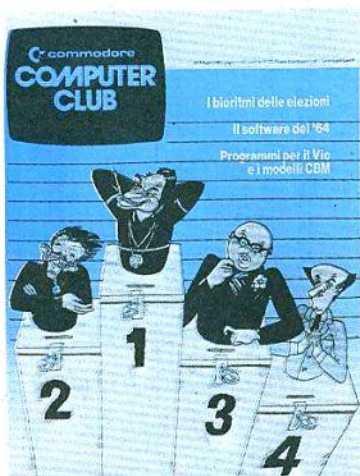
FIG. 1 *Una relazione teorica predatore-preda: una soluzione alle equazioni di Lotke-Volterra*

```
100 REM  PREDATORI E PREDE
110 REM  DI
115 REM  ANTONIO VISCONTI
120 REM  QUALSIASI COMPUTER
125 :
135 REM  SM, CM, NC, J, I DEVONO
    ESSERE
140 REM  CAMBIATI SECONDO IL COMP
    UTER
145 REM  UTILIZZATO
150 :
155 SM=1024:CM=55296:NC=40:REM  C
    -64
165 I=53280:J=53281:REM  C-64
170 :
175 REM  SM=MEMORIA SCHERMO
180 REM  CM=MEMORIA COLORE
185 REM  NC=NUMERO CARATTERI PER
    RIGA
190 REM  I=COLORE BORDO
195 REM  J=COLORE SFONDO
200 :
210 REM  DATI INIZIALI
215 :
220 DIM P(26,41),IN(26,41),M(26,4
    1)
225 POKE  I,14:POKE  J,14
230 PRINTCHR$(144)
235 INPUT "[CLEAR][DOWN]NUMERO DI
     RIGHE   20[4 LEFT]";LC:IF LC
    <0 OR LC>24 THEN 235
240 INPUT "[DOWN]NUMERO DI COLONN
    E   30[4 LEFT]";LR:IF LR<0 OR
     LR>40 THEN 240
245 INPUT "[DOWN] DENSITA' INIZIA
    LE   0.3[5 LEFT]";D:IF D<0 OR
     D>1 THEN 245
250 INPUT "[DOWN]PERCENTUALE INIZ
    IALE PREDATORI     0.2[5 LEFT
    ]";NS:IF NS<0 OR NS>1 THEN 25
    0
255 INPUT "[DOWN]PERIODO DI RIPRO
    DUZIONE PREDE    3 [4 LEFT]";P
    P:PP=PP+10
260 INPUT "[DOWN]PERIODO DI RIPRO
    DUZIONE PREDATORI    6 [4 LEFT
    ]";SP:SP=SP+1000
265 INPUT "[DOWN]INEDIA DEI PREDA
    TORI    3 [4 LEFT]";M
270 PRINT"[CLEAR][3 DOWN]ATTENDER
    E PREGO"
275 :
280 REM  * PREPARA PRIMA CONFIGUR
    AZIONE *
285 :
290 FOR I=0 TO LC+1:P(I,0)=1000:P
    (I,LR+1)=1000:NEXTI
295 FOR J=0 TO LR+1:P(0,J)=1000:P
    (LC+1,J)=1000:NEXTJ
300 FOR I=1 TO LC:FOR J=1 TO LR:A
    =RND(1)
305 IF  A<1-D  THEN  P(I,J)=0:GOT
    O  320
310 A=RND(1):IF A<NS THEN P(I,J)=
    1000+INT(RND(1)*SP+1):GOTO 32
    0
315 P(I,J)=10+INT(RND(1)*PP+1)
320 NEXT J,I
325 PRINT"[CLEAR]"
330 :
335 REM  **      VISUALIZZAZIONE
         **
340 :
345 FOR I=1 TO LC:FOR J=1 TO LR:M
    (I,J)=0:NEXTJ,I
350 XX=0:YY=0:FOR I=1 TO LC:FOR J
    =1 TO LR:K=J+I*NC
355 IF P(I,J)=0 THEN POKE SM-1+K,
    32:GOTO 370
360 IF P(I,J)<1000 THEN POKE SM-1
    +K,46:POKE CM-1+K,1:XX=XX+1:G
    OTO 370
365 POKE SM-1+K,43:POKE CM-1+K,0:
    YY=YY+1:GOTO 370
370 NEXTJ,I
375 PRINT"[HOME]
                 ":REM .30 CARATT
    ERI
380 PRINT"[HOME] PREDE =";XX;"  P
    REDATORI =";YY
385 :
390 REM  ** SCORRIMENTO MATRICE P
    (I,J) PER RILEVARE LE POSIZIO
    NI OCCUPATE **
395 :
400 FOR I=1 TO LC:FOR J=1 TO LR:C
    =0:S=0
405 IF P(I,J)=0 OR M(I,J)<>0 THEN
     495
410 IF P(I-1,J)<>0 THEN C=C+1:S=S
```

```
    +P(I-1,J)
415 IF P(I+1,J)<>0 THEN C=C+2:S=S
    +P(I+1,J)*2
420 IF P(I,J-1)<>0 THEN C=C+4:S=S
    +P(I,J-1)*4
425 IF P(I,J+1)<>0 THEN C=C+8:S=S
    +P(I,J+1)*8
430 IF P(I,J)>=1000 THEN 480
435 :
440 REM  **    SPOSTAMENTO PREDE
         **
445 :
450 ON CGOSUB 520,525,530,535,540
    ,545,550,555,560,565,570,575,
    580,585
455 IF C=0 THEN A%=RND(1)*4+1:ON
    A%GOSUB 590,605,620,635
460 GOTO 495
465 :
470 REM  **    SPOSTAMENTO PREDATO
    RI  **
475 :
480 S=INT(S/1000):IF IN(I,J)>M TH
    EN P(I,J)=0:IN(I,J)=0:GOTO 49
    5
485 ON SGOSUB 665,670,675,680,685
    ,690,695,700,705,710,715,720,
    725,730,735
490 IF S=0 THEN A%=RND(1)*4+1:ON
    A%GOSUB 740,760,780,800
495 NEXT J,I
500 GOTO 345
505 :
510 REM  **    SOTTOPROGRAMMI SPOS
    TAMENTO PREDE    **
515 :
520 A%=RND(1)*3+1:ON A%GOSUB 605,
    620,635:RETURN
525 A%=RND(1)*3+1:ON A%GOSUB 590,
    620,635:RETURN
530 A%=RND(1)*2+1:ON A%GOSUB 620,
    635:RETURN
535 A%=RND(1)*3+1:ON A%GOSUB 590,
    605,635:RETURN
540 A%=RND(1)*2+1:ON A%GOSUB 605,
    635:RETURN
545 A%=RND(1)*2+1:ON A%GOSUB 590,
    635:RETURN
550 GOSUB 635:RETURN
555 A%=RND(1)*3+1:ON A%GOSUB 590,
    605,620:RETURN
560 A%=RND(1)*2+1:ON A%GOSUB 605,
    620:RETURN
565 A%=RND(1)*2+1:ON A%GOSUB 590,
    620:RETURN
570 GOSUB 620:RETURN
575 A%=RND(1)*2+1:ON A%GOSUB 590,
    605:RETURN
580 GOSUB 605:RETURN
585 GOSUB 590:RETURN
590 QQ=I-1
595 IF P(I,J)>PP THEN P(QQ,J)=10:
    P(I,J)=10:RETURN
600 P(QQ,J)=P(I,J)+1:P(I,J)=0:RET
    URN
605 QQ=I+1:M(QQ,J)=1
610 IF P(I,J)>PP THEN P(QQ,J)=10:
    P(I,J)=10:RETURN
615 P(QQ,J)=P(I,J)+1:P(I,J)=0:RET
    URN
620 QQ=J-1
625 IF P(I,J)>PP THEN P(I,QQ)=10:
    P(I,J)=10:RETURN
630 P(I,QQ)=P(I,J)+1:P(I,J)=0:RET
    URN
635 QQ=J+1:M(I,QQ)=1
640 IF P(I,J)>PP THEN P(I,QQ)=10:
    P(I,J)=10:RETURN
645 P(I,QQ)=P(I,J)+1:P(I,J)=0:RET
    URN
650 :
655 REM  **    SOTTOPROGRAMMI SPOS
    TAMENTO PREDATORI   **
660 :
665 A%=RND(1)*3+1:ON A%GOSUB 760,
    780,800:RETURN
670 A%=RND(1)*3+1:ON A%GOSUB 740,
    780,800:RETURN
675 A%=RND(1)*2+1:ON A%GOSUB 780,
    800:RETURN
680 A%=RND(1)*3+1:ON A%GOSUB 740,
    760,800:RETURN
685 A%=RND(1)*2+1:ON A%GOSUB 760,
    800:RETURN
690 A%=RND(1)*2+1:ON A%GOSUB 740,
    800:RETURN
695 GOSUB 800:RETURN
700 A%=RND(1)*3+1:ON A%GOSUB 740,
    760,780:RETURN
705 A%=RND(1)*2+1:ON A%GOSUB 760,
    780:RETURN
710 A%=RND(1)*2+1:ON A%GOSUB 740,
```

```
    780:RETURN
715 GOSUB 780:RETURN
720 A%=RND(1)*2+1:ON A%GOSUB 740,
    760:RETURN
725 GOSUB 760:RETURN
730 GOSUB 740:RETURN
735 IN(I,J)=IN(I,J)+1:RETURN
740 QQ=I-1
745 IF P(I,J)>SP THEN P(QQ,J)=100
    0:P(I,J)=1000:IN(I,J)=IN(I,J)
    +1:RETURN
750 IF P(QQ,J)>0 THEN IN(I,J)=-1
755 P(QQ,J)=P(I,J)+1:P(I,J)=0:IN(
    QQ,J)=IN(I,J)+1:IN(I,J)=0:RET
    URN
760 QQ=I+1:M(QQ,J)=1
765 IF P(I,J)>SP THEN P(QQ,J)=100
    0:P(I,J)=1000:IN(I,J)=IN(I,J)
    +1:RETURN
770 IF P(QQ,J)>0 THEN IN(I,J)=-1
775 P(QQ,J)=P(I,J)+1:P(I,J)=0:IN(
    QQ,J)=IN(I,J)+1:IN(I,J)=0:RET
    URN
780 QQ=J-1
785 IF P(I,J)>SP THEN P(I,QQ)=100
    0:P(I,J)=1000:IN(I,J)=IN(I,J)
    +1:RETURN
790 IF P(I,QQ)>0 THEN IN(I,J)=-1
795 P(I,QQ)=P(I,J)+1:P(I,J)=0:IN(
    I,QQ)=IN(I,J)+1:IN(I,J)=0:RET
    URN
800 QQ=J+1:M(I,QQ)=1
805 IF P(I,J)>SP THEN P(I,QQ)=100
    0:P(I,J)=1000:IN(I,J)=IN(I,J)
    +1:RETURN
810 IF P(I,QQ)>0 THEN IN(I,J)=-1
815 P(I,QQ)=P(I,J)+1:P(I,J)=0:IN(
    I,QQ)=IN(I,J)+1:IN(I,J)=0:RET
    URN
```

Il N.4

Commodore Club, proponendo un videogioco che, come protagonista, ha nientemeno che Craxi! Perfino "La Repubblica", "La Notte" e altri periodici hanno parlato di Craxirunner, indovinato nome del videogioco in oggetto.

La presentazione di un semplice programma di bioritmi è un'ottima scusa per una copertina un pò inconsueta in una rivista di informatica. Il tema dei personaggi del Palazzo è stato ripreso, proprio un paio di settimane fa (ottobre '85), dall'altra rivista su nastro,

QUALSIASI COMPUTER

Variazioni demografiche

Da molti decenni, ormai, il problema della sovrappopolazione preoccupa studiosi (e non) che indicano, per la Terra, presunti limiti massimi consentiti, spesso in contraddizione tra varie teorie.

Si sa, infatti, che le risorse del pianeta non sono indefinite e, prima o poi, occorrerà pensare seriamente ad evitare sprechi e annullare (?) inquinamenti che, come è noto, rappresentano, ai fini dei bilanci ecologici, veri e propri sprechi di ossigeno, acqua, piante, minerali ed altre materie prime in generale che vengono, di fatto, rese inutilizzabili.

Con un piccolo computer non è certo possibile realizzare un modello matematico che, tenendo conto dei numerosi parametri in gioco (diverse migliaia), riesca a fornire con sufficiente approssimazione l'immagine statistica di una "situazione" sociale, politica, umana, economica.

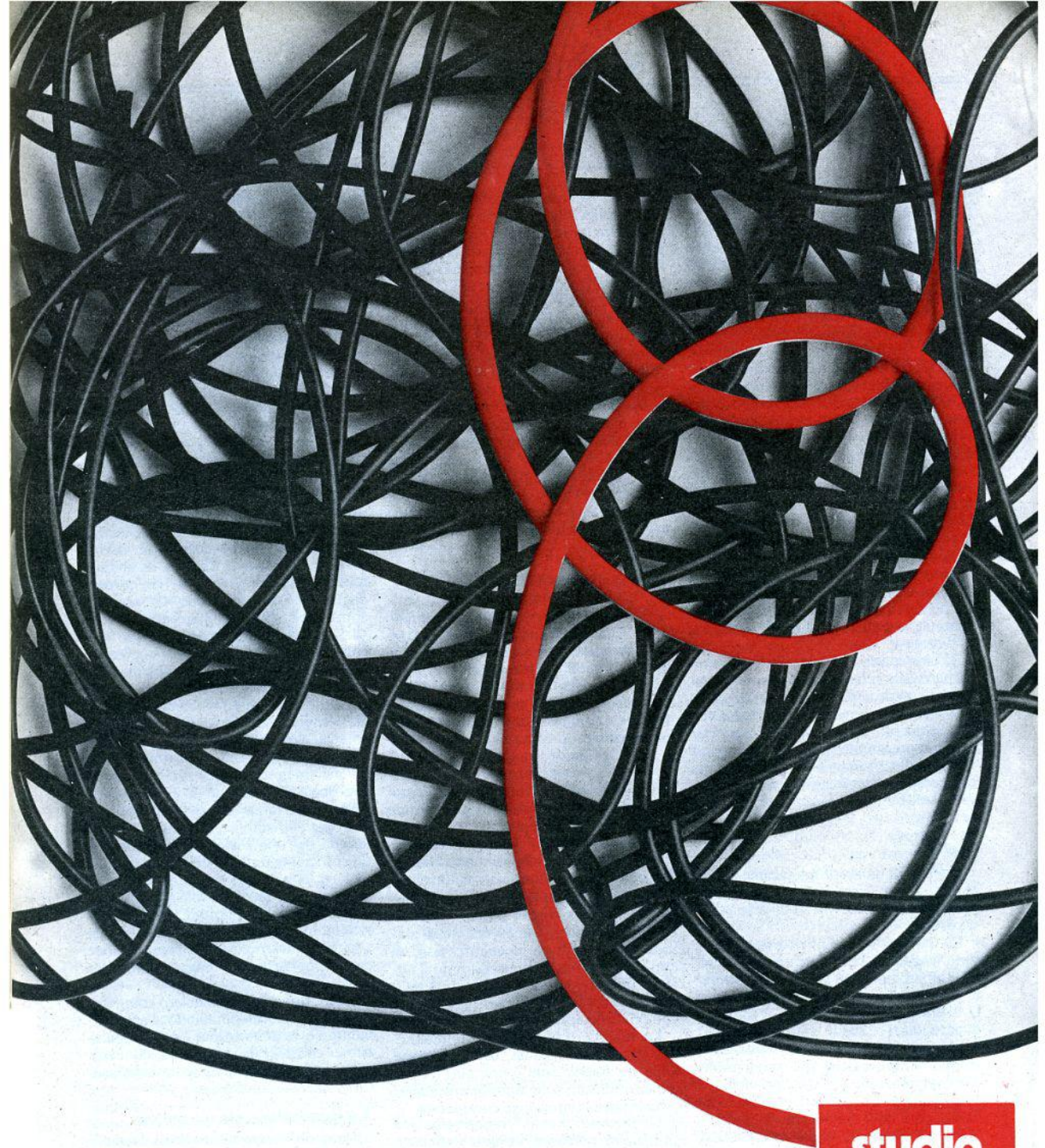
Il micro programma che presentiamo, però, vuole di spronare i più bravi a tentare in tal senso: non è detto che modelli particolarmente interessanti non possano esser riprodotti da un piccolo Commodore e, di conseguenza, pubblicati sulla nostra rivista.

Come realizzare su computer una simulazione qualsiasi (come l'accrescimento della popolazione mondiale), sviluppando, da una semplice idea, una serie di elaborazioni più difficili a dire che a digitare.

Il modello matematico

Per modello matematico si intende quel particolare processo di elaborazione per cui, avendo come presupposto alcune ipotesi convertibili in numeri, si può prevedere l'evoluzione di alcuni fenomeni con un'approssimazione piuttosto precisa, tale da assicurare una certa credibilità alla simulazione stessa.

Se, ad esempio, teniamo conto del peso di un'autovettura, della sua velocità, del tipo di pneumatici che monta e del tipo di manto stradale, è possibile studiare un modello matematico in grado di calcolare lo spazio di frenata ad una certa velocità. E' ovvio che più dettagliato è il modello, più precisa risulta la simulazione. Se non viene considerato il caso della strada bagnata in caso di pioggia oppure non si tiene conto della prontezza di riflessi o del caso di una frenata in curva o su fondo ghiaioso, il modello risulterà meno preciso e di uso meno universale.

In alcuni casi, come quello relativo al calcolo dello spazio di frenata, la simulazione non costituisce grossi problemi, dato che risulta agevole individuare, e

quantificare, tutti gli aspetti del problema.

In altri casi, invece, i fattori che entrano in gioco sono talmente numerosi, e talvolta anche poco chiari, che il modello cui si perviene può essere considerato attendibile solo in una certa percentuale e per fenomeni relativi ad un arco di tempo piuttosto limitato. Come esempio noto a tutti, si pensi al programma AFRODITE col quale il servizio meteorologico dell'aeronautica italiana tenta, ogni giorno, di indovinare il tempo del giorno dopo.

La premessa

Un qualsiasi modello matematico parte da alcuni punti "fermi" ben precisi, che hanno il compito di considerare una certa realtà pur se affetta da un'inevitabile approssimazione.

Non è infatti possibile prevedere, ed inserire nel calcolatore, le migliaia di casi particolari che possono realmente presentarsi nella situazione oggetto dell'ipotesi.

D'altra parte, prendere in considerazione i fattori che possono introdurre variazioni molto piccole all'interno di un quadro generale non può che rallentare l'elaborazione senza, peraltro, apportare un apprezzabile vantaggio rispetto ad un modello più facile e schematico ma che presenta il vantaggio di essere elaborato velocemente e semplice da comprendere.

L'idea

Si è partiti dalla considerazione che gli abitanti della Terra, o di una sua parte, possano esser suddivisi in diverse "classi" di età.

La Statistica, infatti, ricorre spesso a simili espedienti per studiare meglio alcuni fenomeni come gli interessi culturali, il modo di trascorrere il tempo libero, la distribuzione del reddito, il consumo di particolari bevande o altro.

E' necessario evitare suddivisioni troppo semplicistiche (es. suddivisione della popolazione soltanto in bambini e adulti) che possono trarre in inganno alcuni aspetti della simulazione.

Se, ad esempio, vogliamo conoscere l'età di coloro che fumano sigari, il venire a conoscenza che il 100% dei consumatori si colloca presso gli "adulti" piuttosto che presso i "bambini" è certamente un'informazione limitativa.

Suddivisioni troppo raffinate possono egualmente "disperdere" l'attenzione di chi si occupa del problema, vanificando il fine della simulazione stessa.

Se, ad esempio, dividiamo per fasce di età gli allievi che frequentano una scuola superiore (età minima: 13 anni, massima 18), è sufficiente, a seconda del numero di allievi stessi (che supponiamo di almeno un migliaio), suddividerli in una ventina di "classi": quattro trimestri per ciascun anno (per ognuno dei cinque anni) sono idonei per avere un'idea della situazione. Una suddivisione più accurata, come una classe per ogni mese di ciascun anno, potrebbe dare un numero di informazioni "eccessive" col risultato di rendere illegibile la suddivisione. Ciò non toglie, tuttavia, che una suddivisione più accurata, soprattutto in presenza di una popolazione più numerosa (es. studio della distribuzione dell'età di tutti gli studenti italiani), possa mettere in evidenza aspetti altrimenti non rilevabili.

La popolazione terrestre

E' risaputo che il numero di abitanti della Terra (o di una qualunque nazione o... famiglia) varia a seconda di due elementi: la natalità e la mortalità.

Con l'aumentare del numero di figli che ciascuna coppia mette al mondo, cresce il numero di individui della popolazione.

Analogamente, una mortalità infantile ridotta ed una longevità elevata, contribuiscono ad aumentare, in un territorio, il numero di abitanti che vivono "contemporaneamente".

Suddividendo per classi di età la popolazione, ad esempio, di una nazione, possiamo trarre informazioni utilissime.

La tabella pubblicata si riferisce a tre ipotetiche popolazioni (A, B, C) di altrettanti territori i cui 50 milioni di abitanti sono suddivisi in quattro fasce di età: dalla prima, in cui è riportato il numero di individui con età compresa tra 0 e 15 anni, all'ultima (numero di individui con più di 60 anni di età).

La tabella, è bene dirlo, "fotografa" un particolare istante dell'evoluzione delle tre popolazioni e non può fornire indicazioni utili sulle epoche precedenti o successive la situazione considerata.

La popolazione "A" rappresenta una situazione piuttosto insolita (=limite), dato che in ognuna delle classi di età è presente lo stesso numero di individui (10 milioni).

Gli individui la cui età è compresa tra 0 e 15 anni sono piuttosto numerosi nella popolazione "B" (ben 20 milioni), mentre il numero di anziani (più di 60 anni) è addirittura nullo. Si può arguire che la natalità è elevata e che non vi è, nella nazione esaminata, un adeguato sviluppo della medicina o dell'assistenza sociale dato che nessun adulto riece a superare i 60 anni.

Osservando i dati della popolazione "C" (al contrario della "B" che viene considerata "giovane"), possiamo invece notare che il numero di individui giovanissimi è modesto in confronto a quelli di età maggiore. Ciò può esser dovuto ad un'elevata mortalità infantile oppure al fenomeno tipico delle società occidentali che si avvicinano alla cosiddetta "crescita zero" per cui il numero di abitanti tende a rimanere costante col passare degli anni. La popolazione "C", tenendo conto dell'influenza percentuale delle varie classi di età, è da considerersi "vecchia". Questa anzianità della popolazione porta, come è noto, al problema degli anziani che, nei decenni precedenti, non era conosciuto se non in minima misura, dato che la vita media non era elevata come al giorno d'oggi.

In tutti e tre i casi, profondamente diversi tra loro, risulta evidente che la conoscenza del solo totale di abitanti (50 milioni) trarrebbe in inganno il lettore.

Come abbiamo già detto, inoltre, una tabella del tipo esaminato non fornisce un sufficiente numero di informazioni su una certa popolazione, proprio perchè rappresenta i dati di un solo anno di quella particolare popolazione. Interessante risulta, invece, conoscerne la storia, vale a dire la variazione della composizione della popolazione, magari anno per anno.

Ecco, dunque, che il nostro computer può intervenire simulando l'accrescimento (o la diminuzione) della popolazione considerata, partendo da certe premesse supportate da ipotesi che possono essere scelte a piacere.

La simulazione e le sue ipotesi

Il ragionamento su cui si basa il semplicissimo listato riportato in queste pagine

Pop.	0-15	15-30	30-45	45-60	>60	Tot.
A	10	10	10	10	10	50
B	20	25	3	2	0	50
C	7	10	15	10	8	50

inizia dal suddividendo la popolazione nelle quattro fasce di età prima esaminate (vedi figura). Chiunque abbia un po' di pratica, si accorgerà, ad ogni modo, che non vi sono difficoltà di sorta per variare il numero delle fasce di età.

La seconda ipotesi presa in esame consiste nel fatto che, tra le quattro, solo alle due classi centrali (15-30 e 30-45 anni) appartengono, per ovvi motivi, gli individui in grado di mettere al mondo figli.

Si prevede, inoltre, un coefficiente di "sopravvivenza" (il cui valore può variare tra zero ed uno) che può esser diverso per ciascuna classe di età.

La terza considerazione tiene conto del numero di figli che ciascuna coppia può mettere al mondo all'interno del periodo-fascia di età considerato. Per determinare l'accrescimento della popolazione, in base al numero di figli, si è ipotizzato che gli individui di ciascuna classe di età sono per metà maschi e per metà femmine.

Nel listato pubblicato si ipotizza, in altre parole, che se in un certo momento vi sono 32.5 milioni di individui appartenenti alla fascia di età compresa tra 30 e 45 anni, vorrà dire che 16.25 milioni saranno costituiti da maschi e altrettanti da femmine. Tenendo inoltre conto che per fare un figlio è necessaria una femmina e almeno (!) un maschio e supponendo inoltre che, nel periodo di quindici anni considerato (30-45), ogni coppia mette al mondo "in media" un solo figlio, la popolazione risulterà aumentata, sempre nell'intervallo considerato, di 16.25 milioni di individui.

Un'approssimazione apparentemente eccessiva

L'ipotesi appena esaminata può venir considerata inattendibile dato che, come è noto, nessuna delle ipotesi prese in considerazione viene verificata nella realtà.

Non capita mai, infatti, che metà della popolazione sia di sesso maschile e metà femminile (e senza considerare le altre tipologie emergenti!). Non si verifica mai, inoltre, che tutti gli individui si accoppino, nè, tanto meno, che ciascuna coppia metta al mondo sempre e soltanto un solo figlio nel quindicennio considerato e, se ciò non bastasse, metà esatta dei nati appartiene ad un sesso e metà all'altro!

Considerando la simulazione in modo globale, la contraddizione sparisce però immediatamente. Analizziamo a tale scopo, per esemplicità, una sola fascia di età (30-45) e paragoniamo fra loro due ipotesi: quella apparentemente contraddittoria ed un'altra, al contrario, più realistica.

Prima ipotesi

Popolazione: 32.5 milioni di persone appartenenti alla stessa fascia di età.
Maschi: 16.25 milioni
Femmine: 16.25 milioni
Coppie capaci di generare figli: 16.25 milioni
N. figli generati da ciascuna coppia nel quindicennio: uno solo
Situazione alla fine del quindicennio:

Totale popolazione adulta: 32.5. Maschi: 16.25. Femmine: 16.25. Figli: 16.25

Seconda ipotesi

Popolazione: 32.5 milioni
Maschi: 12.5 milioni
Femmine: 20 milioni
Coppie capaci di generare figli: 10.22 milioni (vi sono, cioè, anche nubili e celibi oltre che coppie)
N. di coppie che non mettono al mondo figli: 1 milione
N. di coppie che generano un solo figlio nel quindicennio considerato: 4.8 milioni
N. coppie che generano due figli (nello stesso periodo): 2.12 milioni
N. di coppie che generano tre figli: 2.4 milioni
Situazione alla fine del quindicennio considerato:

Totale popolazione adulta: 32.5 milioni
Figli generati: 16.24 (=0 + 4.8*1 + 2.12*2 +2.4*3)

Come si può notare il risultato finale è praticamente identico nelle due ipotesi pur se queste sono sensibilmente diverse tra loro.

Come funziona il programma

Si parte, come già detto, dall'ipotesi che la popolazione viene suddivisa in quattro fasce di età assegnate agli elementi del vettore A(i). Un'altra premessa consiste nell'assegnare, per semplicità, lo stesso valore (100) al numero di individui di ciascuna fascia. Si parte, in altre parole, con una popolazione di 400 unità suddivisa in quattro settori.

100 può essere indifferentemente il numero di centinaia, di migliaia, di milioni, di miliardi o di... semplici individui da considerare. E' superfluo ricordare che è possibile assegnare un valore completamente diverso sia in assoluto sia per classe apportando le modifiche del caso alla riga 1060.

Viene, in seguito, considerata la "sopravvivenza" (SOPR.) di ciascuna fascia. Per sopravvivenza si intende un coefficiente Z(i), che può esser diverso da fascia a fascia, compreso tra zero ed uno.

Il valore unitario indicherà che il 100% della popolazione della fascia considerata sopravvive fino alla fascia successiva. Una valore 0.83 indicherà al programma che solo l'83% della popolazione sopravvive mentre il rimanente 17% perirà. Altri esempi:
0.7=70%
0.1=10%
0.31=31% ecc.

Un'altra ipotesi che è facile riscontrare, consiste nel limitare alla seconda (15-30) e terza fascia (30-45) la possibilità di generare figli. La possibilità di generare individui da parte della popolazione compresa tra 0 e 15 anni, come pure quella oltre i 45 anni, si considera ininfluente ai fini della simulazione.

Facendo ora riferimento alla figura allegata si può facilmente notare che, al termine di ogni quindicennio considerato, si verificano i seguenti fenomeni:

- la popolazione della prima (seconda, terza...) fascia è cresciuta di 15 anni e di conseguenza farà parte, nel ciclo successivo, della seconda (terza, quarta...) fascia tenendo conto, ovviamente, della sopravvivenza impostata precedentemente;
- nel quindicennio considerato le coppie appartenenti alla seconda e terza fascia hanno generato un certo numero di figli, i quali appartengono tutti indistintamente alla prima fascia. Questa, anzi, può esistere solo grazie ai figli generati. Se, infatti, si suppone che il numero di figli generati sia sempre nullo, l'intera popolazione perirebbe nel giro di 60 anni!

Il numero di figli messi al mondo è dato dal numero di coppie della fascia considerata moltiplicato per il numero di figli che, mediamente, ciascuna coppia mette al mondo. Il numero di coppie è ovviamente dato dal numero di individui della fascia diviso per due. Per superare tale eccessiva semplificazione, riscontrata precedentemente, è possibile agire sul numero di figli che ciascuna coppia può generare.

Non sono, pertanto, privi di significato valori di 1.7 oppure 2.3 o altri valori

rappresentanti il numero "medio" di figli per coppia "media".

Il numero di individui che supera il limite dei 60 anni non viene considerato (non ce ne vogliano i lettori meno giovani!) ma, come già detto, sarà semplicissimo aggiungere fasce anche di ultracentenari.

Come gira il programma

Dato il consueto RUN, appariranno alcune domande che ora esaminiamo:
SOPRAVV. FASCIA 1?

Rispondere con 1 se la mortalità infantile è ipotizzata nulla. In caso contario digitare un valore non minore di zero e non maggiore di uno. (es.: 0.80 0.92 eccetera)

Comportarsi allo stesso modo alla richiesta della sopravvivenza delle altre tre fasce di età.
N. FIGLI FASCIA 2?

Rispondere tenendo conto delle considerazioni fatte precedentemente. (es.: 2 1.8 3.2 eccetera)

In modo analogo rispondere alla successiva domanda.

Quando compare: N. CICLI?

Indicare il numero di simulazioni che devono essere effettuate tenendo fissi i parametri impostati con le risposte precedenti. (es.: 3)

A questo punto ogni volta che si preme un tasto verrà effettuata una simulazione, vale a dire verrà visualizzato il numero di individui presenti in ciascuna classe di età al termine di un periodo di quindici anni.

Una volta esaurito il numero di simulazioni indicate (NC), il cursore si posizionerà automaticamente in corrispondenza della prima domanda effettuata. A questo punto potrete battere semplicemente il tasto Return nei casi in cui si desidera lasciare inalterata qualcuna delle ipotesi impostate precedentemente. In caso contrario digitate le variazioni che riterrete più opportune.

Nel caso in cui il totale (variabile TT) degli abitanti assuma il valore nullo (ttutti morti!), il programma si ferma automaticamente.

Per i più bravi

Il programma si presta, data la semplice linearità, a sofisticazioni non indifferenti.

Chi vuol tasformare il semplice listato in un vero e proprio modello matematico può, tra l'alto, inserire numerose modifiche in modo che vengano prese in esame, durante l'elaborazione, varie situazioni.

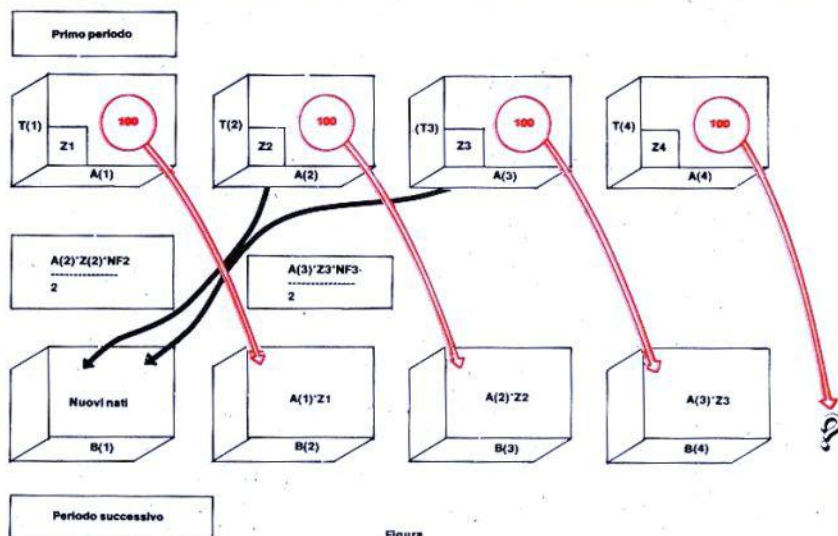
Ecco, per gradi di difficoltà, alcuni suggeriemnti di sofisticazioni possibili:
- Considerare più fasce di età.
- Inserire la memorizzazione dei risultati parziali in matrici che, richiamate in modo opportuno, visualizzino in seguito tutte le simulazioni effettuate.
- Visualizzare, al fianco del numero di individui di ciascuna classe, la percentuale della classe stessa rispetto al totale della popolazione.
- Inserire, in modo Random o sistematico, un evento "distruttore" (guerre, carestie, insufficienza di produzione alimentare) non appena la popolazione supera un certo valore globale.
- Inserire una riduzione automatica della eventuale mortalità infantile in seguito ad ipotetiche migliorie del tenore di vita.
- Calcolare il numero di adulti dediti all'agricoltura e considerare un'estensione credibile di aree coltivabili in modo da verificare la possibilità di nutrire la popolazione.

Alessandro de Simone

... E tante altre idee

Come si può facilmente osservare le modifiche da apportare sono, praticamente, infinite.

I lettori che realizzeranno simulazioni di qualsiasi altra natura, soprattutto se semplicissime come quella proposta, sono pregati di mettersi in contatto telefonicamente con la Redazione allo scopo di concordarne la pubblicazione. Il numero di telefono è sempre il solito: 02/8467348 (meglio se al venerdì)

Figura

```
1000 REM  SEMPLICI SIMULAZIONI
1010 REM  PER QUALSIASI COMPUTER
1015 :
1020 REM  INCREMENTO DEMOGRAFICO
1030 :
1040 REM  A. DE SIMONE
1050 :
1060 A(1)=100:A(2)=100:A(3)=100:A(4)=100
1070 PRINTCHR$(147);:XX$="":FOR I=1 TO 8:XX$=XX$+CHR$(17):NEXT
1080 XX$=CHR$(19)+XX$:XT=0:FOR I=1 TO 8:SP$=SP$+CHR$(32):NEXT
1090 PRINTCHR$(19);:INPUT "SOPR.FASCIA 1";Z(1)
1100 INPUT "SOPR.FASCIA 2";Z(2)
1110 INPUT "SOPR.FASCIA 3";Z(3)
1120 INPUT "SOPR.FASCIA 4";Z(4)
1130 INPUT "N.FIGLI FASCIA 2";N2
1140 INPUT "N.FIGLI FASCIA 3";N3
1150 INPUT "N.CICLI";NC
1160 FOR X1=1 TO NC
1170 B(1)=(A(2)*Z(2)*N2/2)+(A(3)*Z(3)*N3/2)
1180 B(2)=A(1)*Z(1)
1190 B(3)=A(2)*Z(2)
1200 B(4)=A(3)*Z(3)
1210 TT=0:FOR I=1 TO 4:TT=TT+B(I):NEXT
1220 PRINTXX$:FOR I=1 TO 4:PRINT"FASCIA"I;B(I)CHR$(157);:REM  (PUNTO E VIRGOLA)
1230 REM  PERCENTUALE: PRINTB(I)*100/TT
1240 PRINTSP$:NEXTI
1250 FOR I=1 TO 4:A(I)=B(I):NEXTI:XT=XT+1
1260 PRINTCHR$(17)CHR$(18)"TOT.="TTCHR$(157)SP$" N.CICLI=";XTCHR$(157)SP$
1270 PRINT"PREMI UN TASTO"
1275 GET A$:IF A$="" THEN 1275
1280 IF TT THEN NEXTX1:GOTO 1090
```

Effetti sonori

Beep... rumble... squash... e tanti altri effetti speciali per i vostri giochi.

Il SID, ovvero tormento e delizia per le nostre orecchie, è il microprocessore adibito a gestire il suono del C-64.

Tralascio la solita premessa atta a magnificare la potenza e versatilità di questo chip, doti peraltro assai reclamizzate e conosciute da parte degli aficionados dei prodotti Commodore.

Bisogna però riconoscere che la fama di computer "musicoso" se l'è ampiamente meritata e costruita non solo grazie al battage pubblicitario, ma anche proprio per le sue qualità intrinseche sull'incredibile Commodore 64. Non corro certo il rischio di apparire plagiato dai media televisivi facendo tali affermazioni e, senza scendere al livello del tizio che, mentre mangiucchia una caramella balsamica dice: "mia suocera aveva ragione!", bisogna dare merito alla casa americana di aver mantenuto appieno le promesse fatte.

Ma mi fermo qui, altrimenti mi lascerei prendere dall'entusiasmo e, contrariamente a quanto mi ero promesso, vi angustierei con tante chiacchere parlandovi di effetti galattici... anzi mostruosi... cosa dico, esplosivi...

Come usare le routine

Non vogliamo neppure fare l'ennesima lezione su come sfruttare appieno le capacità sonore del nostro personal.

Agli affezionati lettori il menù del giorno di C.C.C. propone: antipasti leggeri, stuzzicanti e facili da digerire.

Tante ricettine semplici e pronte per l'uso; le potrete inserire così come sono nei vostri "piatti forti", oppure spizzicare di tanto in tanto per deliziare i vostri timpani.

I programmi sono tutti ampiamente REMmati e, ove possibile, si è cercato di evitare i comandi di salto (quali GOTO, GOSUB...) affinchè non abbiate problemi nel renumerarli.

Infatti il fine dei listati proposti è appunto quello di essere inseriti nei vostri programmi: basta mettere un RETURN come ultima riga e potranno essere richiamati in qualsiasi momento tramite un GOSUB.

I programmi sono costruiti in maniera tale da poter essere modificati con facilità, per avere la possibilità di ottenere gli effetti desiderati.

Se infatti, nonostante tutta la mia buona volontà, il boing o il rumble non saranno di vostro gradimento, li potete prendere, tagliare, spezzettare, agitare e incollare.

Se però il "vostro" elicottero non è più un elicottero ma è diventato una zanzara, allora... non prendetevela con gli altri (cioè C.C.C.).

Esiste però la possibilità che voi siate tipi dall'ottimismo incrollabile, di quelli, per intenderci, che davanti ad un pacchetto con dentro dieci sigarette dicono che è ancora mezzo pieno. Allora saprete certamente cogliere il lato positivo della faccenda ed apprezzare l'inatteso risultato.

I registri del sid

Giusto perchè voi abbiate sottomano i "ferri del mestiere" (fatene buon uso, non distruggete le mie fatiche...), viene riportato un sintetico specchietto sui registri utilizzati dal SID.

Per indicare i vari indirizzi si farà riferimento alla variabile SI, che tutti i musimaniaci conosceranno a memoria: SI=54272

Flavio Molinari

SI = 54272	VOCE 1	VOCE 2	VOCE 3
FREQUENZA BYTE BASSO (0-255)	SI	SI+7	SI+14
FREQUENZA BYTE ALTO (0-255)	SI+1	SI+8	SI+15
TOCCO SIMULATO ONDA RETTANGOLARE BYTE BASSO (0-255)	SI+2	SI+9	SI+16
TOCCO SIMULATO ONDA RETTANGOLARE BYTE ALTO (0-255)	SI+3	SI+10	SI+17
FORMA D'ONDA 17 TRIANGOLARE 33 DENTE DI SEGA 65 RETTANGOLARE 129 RUMORE BIANCO	SI+4	SI+11	SI+18
ATTACK / DECAY (0-15) (0-15) 16*AT + DE	SI+5	SI+12	SI+19
SUSTAIN / RELEASE (0-15) (0-15) 16*SU + RE	SI+6	SI+13	SI+20

REGISTRI COMUNI ALLE TRE VOCI	
SI+21	FREQUENZA DI TAGLIO BYTE BASSO (0-7)
SI+22	FREQUENZA DI TAGLIO BYTE ALTO (0-255)
SI+23	FILTRI V. (0-15) RISONANZA (0-15)*16
SI+24	VOLUME (0-15) FILTRI (0-15)*16

```
100 REM  *************************
110 REM  *        SIRENA         *
120 REM  *************************
130 :
140 SI=54272
150 FOR  Q=0  TO  24
160 POKE  SI+Q,0:NEXT
170 POKE  SI+5,255:REM  ATTACK/DEC
    AY
180 POKE  SI+24,15:REM  VOLUME
190 POKE  SI+3,14:REM  TOCCO SIMUL
    ATO
200 A1=55:B1=42
210 POKE  SI+1,60
220 POKE SI+4,65
230 FOR  K=1  TO  14:REM  N. CICLI
     SIRENA
240 POKE  SI+1,A1
250 FOR  Q=1  TO  400:NEXT
260 POKE  SI+1,B1
270 FOR  Q=1  TO  400:NEXT
280 IF  K>7  THEN  POKE  SI+24,21-
    K:REM  SIRENA CHE SI ALLONTANA
290 IF  K=7  THEN  A1=54:B1=41
300 IF  K=10  THEN  A1=53:B1=40
310 NEXT
320 POKE  SI+4,0
330 POKE  SI+24,0
500 REM  *************************
510 REM  *        RAZZO          *
520 REM  *************************
530 :
540 SI=54272
550 FOR  Q=0  TO  24
560 POKE  SI+Q,0:NEXT
570 POKE  SI+24,15
580 FR=600:REM  FREQUENZA
590 POKE  SI+1,INT(FR/256)
600 POKE  SI,FR-256*INT(FR/256)
610 POKE  SI+5,0
620 POKE  SI+6,255
630 POKE  SI+4,129:REM  RUMORE
640 FOR  K=250  TO  1  STEP -1:REM
      DURATA DECOLLO
650 FR=FR+100:REM  AUMENTA LA FREQ
    UENZA
660 IF  K<45  THEN  POKE  SI+24,K/
    3:REM  ABBASSA IL VOLUME
670 S2=INT(FR/256):S1=FR-256*S2
680 POKE  SI,S1
690 POKE  SI+1,S2
700 NEXT
710 POKE  SI+4,0
1000 REM  *************************
1010 REM  *       TELEFONO        *
1020 REM  *************************
1030 :
1040 SI=54272
1050 FOR  Q=0  TO  24:REM  RESET
1060 POKE  SI+Q,0:NEXT
1070 POKE  SI+6,255:REM  SU=16:RE=1
     5
1080 POKE  SI+3,5:REM  TOCCO SIMULA
     TO
1090 POKE  SI+4,65:REM  ONDA RETTAN
     GOLARE
1100 POKE  SI+24,15
1110 FOR  H=1  TO  4:REM  NUMERO TR
     ILLI
1120 FOR  K=1  TO  50:REM  TRILLO
1130 POKE  SI+1,46
1140 FOR  Q=1  TO  5:NEXT:REM  PAUS
     A
1150 POKE  SI+1,0
1160 NEXT
1170 FOR  Q=1  TO  3000:NEXT
1180 NEXT
1190 POKE  SI+4,0
1500 REM  *************************
1510 REM  *        LASER          *
1520 REM  *************************
1530 :
1540 SI=54272
1550 FOR  Q=0  TO  24
1560 POKE  SI+Q,0:NEXT
1570 VL=12:REM  VELOCITA' LASER
1580 POKE  SI+24,15
1590 POKE  SI+6,240
1600 FOR  K=1  TO  2:REM  NUMERO RA
     GGI
1610 POKE  SI+4,65
1620 POKE  SI+3,5
1630 FOR  T=100  TO  250  STEP  VL
1640 POKE  SI,T
1650 POKE  SI+1,250-T:NEXT
1660 POKE  SI+4,129
1670 POKE  SI+1,2
1680 FOR  Q=1  TO  100:NEXT
```

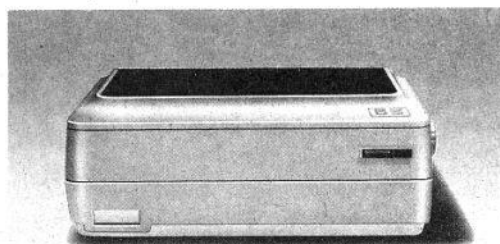
```
1690 POKE  SI+4,0
1700 FOR  Q=1  TO  50:NEXT
1710 NEXT
2000 REM  *************************
2010 REM  *      AEREO 1          *
2020 REM  *************************
2030 :
2040 SI=54272
2050 FOR  Q=0  TO  24
2060 POKE  SI+Q,0:NEXT
2070 POKE  SI+1,2
2080 POKE  SI+3,4
2090 POKE  SI+6,240
2100 POKE  SI+7,12
2110 POKE  SI+8,10
2120 POKE  SI+10,5
2130 POKE  SI+13,192
2140 POKE  SI+14,16
2150 POKE  SI+15,2
2160 POKE  SI+17,6
2170 POKE  SI+20,64
2180 POKE  SI+22,30
2190 POKE  SI+23,243
2200 POKE  SI+24,31
2210 POKE  SI+4,65
2220 POKE  SI+11,65
2230 POKE  SI+18,65
2500 REM  *************************
2510 REM  *       MANIAC          *
2520 REM  *************************
2530 :
2540 SI=54272
2550 FOR  Q=0  TO  24:REM  RESET SU
     ONO
2560 POKE  SI+Q,0:NEXT
2570 POKE  SI+6,255:REM  ATTACK/DEC
     AY
2580 POKE  SI+4,17:REM  TRIANGOLARE
2590 POKE  SI+24,15
2600 FOR  K=1  TO  100:REM  NUMERO
     CICLI
2610 POKE  SI+1,RND(1)*256:REM  FRE
     QUENZA CASUALE
2620 FOR  Q=1  TO  65:NEXT
2630 NEXT
2640 POKE SI+4,0:POKE SI+24,0
3000 REM  *************************
3010 REM  *      ALLARME          *
3020 REM  *************************
3030 :
3040 SI=54272
3050 FOR  Q=0  TO  24
3060 POKE  SI+Q,0:NEXT
3070 POKE  SI+6,240
3080 POKE  SI+3,5
3090 POKE  SI+4,65
3100 POKE  SI+24,15
3110 FOR  K=1  TO  12:REM  NUMERO A
     LLARMI
3120 FOR  H=40  TO  100  STEP  2
3130 POKE  SI+1,H:REM  FREQUENZA
3140 FOR  Q=1  TO  10:NEXT
3150 NEXT
3160 POKE  SI+1,0
3170 FOR  Q=1  TO  100:NEXT:REM  PA
     USA
3180 NEXT
3190 POKE  SI+4,0
3500 REM  *************************
3510 REM  *      ELICOTTERO       *
3520 REM  *************************
3530 :
3540 SI=54272
3550 FOR  Q=0  TO  24
3560 POKE  SI+Q,0:NEXT
3570 POKE  SI+24,15
3580 POKE  SI+1,15:REM  FREQUENZA
3590 POKE  SI+6,7*16:REM  SU=7:RE=0
3600 FOR Q1=1  TO  90: POKE  SI+4,1
     29:REM  RUMORE
3610 FOR  Q=1  TO  20:NEXT:REM  GIR
     O MOTORE
3620 POKE  SI+4,0
3630 FOR  Q=1  TO  50:NEXT:REM  PAU
     SA
3640 NEXT Q1
4000 REM  *************************
4010 REM  *         MARE          *
4020 REM  *************************
4030 :
4040 SI=54272
4050 FOR  Q=0  TO  24
4060 POKE  SI+Q,0:NEXT
4070 POKE  SI+6,255
4080 POKE  SI+5,0
4090 POKE  SI+4,129:REM  RUMORE
4100 POKE  SI+1,40:REM  FREQUENZA
4110 FOR  K=1  TO  8:REM  NUMERO ON
```

```
    DE
4120 DU=RND(1)*250+20:REM  DURATA O
    NDA
4130 FOR  VO=3  TO  15:REM  ONDA CH
    E SALE
4140 POKE  SI+24,VO
4150 FOR  Q=1  TO  DU:NEXT
4160 NEXT
4170 FOR  VO=15  TO  3  STEP -1:REM
      ONDA CHE SCENDE
4180 POKE  SI+24,VO
4190 FOR  Q=1  TO  DU:NEXT
4200 NEXT:NEXT
4210 POKE  SI+4,0
4220
5000 REM  *************************
5000 REM  *************************
5010 REM  *       GNAM-GNAM       *
5020 REM  *************************
5030 :
5040 SI=54272
5050 FOR  Q=0  TO  24
5060 POKE  SI+Q,0:NEXT
5070 POKE  SI+6,255:REM  S/R VOCE 1
5080 POKE  SI+13,255:REM  S/R VOCE
    2
5090 POKE  SI+4,129:REM  RUMORE
5100 POKE  SI+11,65:REM  RETTANGOLA
    RE
5110 POKE  SI+10,2:REM  TOCCO SIMUL
    ATO VOCE 2
5120 POKE  SI+24,15
5130 FOR  H=1  TO  2:REM  NUMERO GN
    AM
5140 FOR  K=255  TO  150  STEP  -10
5150 POKE  SI+8,255-K
5160 POKE  SI+1,K
5170 NEXT:NEXT
5180 POKE  SI+4,0:POKE  SI+11,0
5500 REM  *************************
5510 REM  *         BOMB          *
5520 REM  *************************

5530 :
5540 SI=54272
5550 FOR  Q=0  TO  24
5560 POKE  SI+Q,0:NEXT
5570 POKE  SI+6,240:REM  S/R VOCE 1
5580 POKE  SI+13,255:REM  S/R VOCE
    2
5590 POKE  SI+4,17 :REM  TRIANGOLAR
    E
5600 POKE  SI+8,4
5610 POKE  SI+24,15
5620 FOR  K=255  TO  150  STEP  -.4
    :REM  SIBILO
5630 POKE  SI+1,K
5640 NEXT:POKE  SI+4,0
5650 POKE  SI+11,129
5660 FOR  Q=1  TO  15  STEP  .1:REM
      TONFO
5670 POKE  SI+24,15-Q
5680 NEXT
5690 POKE  SI+4,0:POKE  SI+11,0
```

Le proposte del n. 1

PER STAMPARE CON POCHE LIRE

Le stampanti **MT/85, a 80 colonne, e MT/86, a 136 colonne,** rappresentano una nuova frontiera nel settore delle stampanti a basso costo.
Basso costo, ma non bassa qualità e basse prestazioni, infatti ecco le credenziali di questi due nuovi prodotti.
Velocità a 180 cps. bidirezionale ottimizzata, NLQ a 45 cps., grafiche, possibilità di 8 fonti alternative di caratteri e naturalmente la completa compatibilità con il PC IBM.
Il prezzo: il più competitivo del mercato in questa fascia di prestazioni.
Naturalmente anche le MT/85/86 oltre ai trattori hanno anche trascinamento a frizione e consentono pertanto il trattamento del foglio singolo.

20094 Corsico (MI) - Via Cadamosto, 3
Tel. (02) 4502850/855/860/865/870 - Telex 311371 Tally I
00137 Roma - Via I. Del Lungo, 42 - Tel. (06) 8278458
10099 San Mauro (TO) - Via Casale, 308 - Tel. (011) 8225171
40050 Monteveglio (BO) - Via Einstein, 5 - Tel. (051) 832508

Un percorso probabile

Vi sarà capitato più di una volta di osservare alla televisione o sui banchi di scuola lo svolgimento di un semplice esperimento che consiste nel far cadere un certo numero di palline parallelamente ad una superficie sulla quale sono piantati dei chiodi disposti a triangolo.

Le palline, al termine della prova, si dispongono in modo da rappesentare a loro volta una figura molto simile allo stesso triangolo.

In realtà ciascuna pallina, cadendo, urta contro uno dei chiodi e, per proseguire, è costretta a spostarsi a destra oppure a sinistra incontrando nuovamente un chiodo fino a che non esce dal triangolo.

Se il numero delle palline, e quello dei chiodi costituenti il labirinto, è elevato, le palline stesse, dopo l'esperimento, si disporranno secondo una curva (figura 1) che indica la probabilità di presenza di una generica pallina in un dato punto dopo la caduta.

Come si può verificare, tale probabilità è massima in corrispondenza del punto verticale (al vertice del triangolo) mentre è praticamente nulla in corrispondenza dei lati della base.

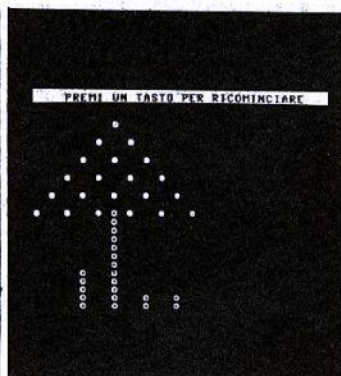
In effetti la pallina, urtando dapprima il chiodo posto al vertice, ha le stesse probabilità di proseguire a destra o a sinistra.

Se, ad esempio, cade a destra, colpirà un secondo chiodo posto diagonalmente a destra rispetto al primo.

La pallina quindi rimbalzerà una seconda volta ed avrà ancora le stesse probabilità di cadere a destra o a sinistra.

In tal modo la pallina, cadendo verso la base del triangolo, segue un percorso casuale.

Per far sì che la pallina cada verso la zona periferica della base, dovrebbe cadere sempre a sinistra oppure sempre a destra ad ogni ostacolo incontrato, e questo evento, naturalmente, è piuttosto improbabile.

E' più probabile, al contrario, che la pallina cada lo stesso numero di volte a destra e a sinistra degli ostacoli pervenendo, alla fine, nella zona centrale della base.

Ecco come simulare, con il computer, la traiettoria seguita da una pallina in un labirinto. Il tutto con poche linee di programma BASIC.

Perchè l'esperimento sia rigoroso si deve però tenere conto anche del fattc che, una volta acquistata una componente di moto, verso destra o verso sinistra la pallina tenderà a conservare questa quantità di moto continuando a rimbalzare verso la stessa direzione.

Per ovviare a questo incoveniete i chiodi dovranno essere posizionati con molta cura.

Naturalmente questo problema non si pone con la simulazione al computer, poichè la pallina è rappresentata da un carattere grafico e come tale non possiede nè massa nè tantomeno una quantità di moto.

Il programma

Il programma non è per nulla complicato: eliminate le linee di commento si riduce a qualcosa di estremamente breve.

I caratteri grafici corrispondenti alle palline vengono stampati attraverso un'istruzione di POKE, mentre la presenza di un ostacolo è rivelata con una PEEK.

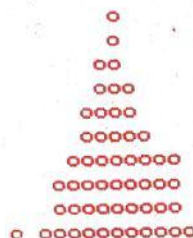
Bisogna dire che non sempre le palline cadute riproducono alla perfezione l'immagine del triangolo di "chiodi", a causa del numero limitato di palline cadute e di ostacoli presenti sul loro cammino.

Perchè l'esperimento riesca alla perfezione i "chiodi" dovrebbero essere molti di più e così anche le palline cadute, ma le dimensioni dello schermo, purtroppo, non lo permettono.

La maggior parte delle volte, comunque, la curva viene rappresentata bene e si può notare come nelle zone estreme a volte non cada nemmeno una pallina, mentre nelle zone centrali si accumulano in gran numero.

Luca Galuzzi

```
100 REM  UN PERCORSO PROBABILE
110 :
130 REM  SIMULAZIONE DI L. GALUZZI
140 :
150 REM  VIC 20 . C-64
160 REM  C-16 . PLUS 4
170 :
180 :
190 PRINTCHR$(147)"1- COMMODORE 64
    "
200 PRINT"2- VIC 20 INESPANSO"
210 PRINT"3- C-16"
220 GET A$:IF A$="" THEN 220
230 X1=1024:X2=40:X4=23
240 IF A$="2" THEN X1=7680:X2=22:X
    4=21
250 IF A$="3" THEN X1=3072
260 REM  *
       *
270 REM  *************************
    ****
280 PRINT CHR$(147)CHR$(17):XA$=CH
    R$(29)+CHR$(29)+CHR$(29)
290 PRINT  TAB(10)CHR$(17) "●":REM
     SHIFT + "Q"
300 PRINT  TAB(8) CHR$(17)"●"XA$"●
    "
310 PRINT  TAB(6) CHR$(17)"●"XA$"●
    "XA$"●"
320 PRINT  TAB(4) CHR$(17)"●"XA$"●
    "XA$"●"XA$"●"
330 PRINT  TAB(2) CHR$(17)"●"XA$"●
    "XA$"●"XA$"●"XA$"●"
340 PRINT  CHR$(17)"●"XA$"●"XA$"●"
    XA$"●"XA$"●"XA$"●"
350 X=10:Y=0:GOSUB  430:PRINT "o";
    :REM  SHIFT + "W"
360 Y=Y+1
370 PRINT CHR$(157)CHR$(32);
380 GOSUB  430:PRINT "o";
390 IF  PEEK(X1+(X2*(Y+2))+X)=81
    THEN  GOSUB  500
400 IF  PEEK(X1+(X2*(Y+2))+X)=87
    THEN  550
410 IF  Y=X4  THEN  350
420 GOTO  360
430 REM  *************************
    *
440 REM  ***   PRINT AT (X,Y)    **
    *
450 REM  *************************
    *
460 PRINT CHR$(19)
470 FOR  I=1  TO  Y:PRINT CHR$(17)
    ;:NEXT
480 FOR  I=1  TO  X:PRINT CHR$(29)
    ;:NEXT
490 RETURN
500 REM  *************************
    *
510 REM  ***       URTO           **
    *
520 REM  *************************
    *
530 IF  (INT(RND(1)*2))=1  THEN  X
    =X-2:RETURN
540 X=X+2:RETURN
550 IF  Y<=12  THEN  570
560 GOTO  350
570 PRINT CHR$(19)CHR$(18)"     PRE
    MI UN TASTO PER RICOMINCIARE
     "
580 GET A$:IF  A$=""  THEN  580
590 GOTO  280
```

Chiodi

Palline

Caratteri al microscopio

Come esaminare, vistosamente ingranditi su video oppure su stampante, i caratteri del popolare computer simulando l'uso di un microscopio puntato sul teleschermo.

Quando con una normale macchina da scrivere battiamo una lettera possiamo vedere (e capire) come i caratteri vengono stampati.

Quando invece usiamo il computer, i caratteri che digitiamo da tastiera compaiono, come per magia, sul video.

Questa magia, detta elettronica, nel bene o nel male ci accompagnerà per i secoli a venire...

Ora, tornando alla nostra vecchia macchina da scrivere, sappiamo che esistono dispositivi meccanici (di vario tipo) che contengono tutti i caratteri alfanumerici che consentono di scrivere.

Se anche col computer riusciamo a fare tutto ciò significa che anche questo strumento possiede un dispositivo contenente tutti i caratteri utili.

Questo dispositivo, che evidentemente non è di tipo meccanico, è detto ROM GENERATORE DI CARATTERI.

*** Il termine ROM (Read Only Memory) significa Memoria a Sola Lettura: è una parte, cioè, della memoria del computer da cui noi possiamo solo leggere dati, senza la possibilità di modificarne il contenuto.

Per contro, la parte di memoria dove possiamo anche scrivere è detta RAM (Random Access Memory = Memoria ad Accesso Casuale): questa parte di memoria è quella in cui scriviamo i nostri programmi, le variabili, dati eccetera.***

Dunque, ora sappiamo che tutti i nostri caratteri si trovano in una zona della memoria detta Rom Generatore di Caratteri.

Vediamo ora, in dettaglio, come ogni singolo carattere è costruito all'interno della macchina.

Tanto per cominciare, si potrebbe osservare attentamente una qualsiasi lettera sullo schermo: se il video è abbastanza grande (e la vista buona!), ogni carattere ci apparirà come un serie di punti, disposti in modo da creare la lettera osservata.

Quindi, la conclusione che possiamo trarre è che i caratteri sono composti da più punti "accesi" sullo schermo in contrasto con altri che sono, invece, "spenti".

I caratteri, in effetti, sono formati da una matrice (griglia) fatta da 8 8 punti (8 righe 8 colonne).

Ogni riga, formata da 8 punti, è detta BYTE; ogni punto, invece, è detto BIT: segue che un byte è formato da 8 bit.

Il bit, è l'UNITA' ELEMENTARE DEL COMPUTER e, convenzionalmente, può assumere solo due valori: 0 o 1.

Pensate ad un interruttore: questo può essere spento, 0, oppure acceso, 1.

Tutti i bit con valore 1 costituiscono i punti accesi (ON) sul video, mentre quelli con valore 0 i punti spenti (OFF).

*** I singoli punti accesi sullo schermo sono detti PIXEL e costituiscono le UNITA' FONDAMENTALI dell'immagine video: punti più piccoli non possono essere accesi!***

L'insieme degli 8 byte, con i relativi bit accesi o spenti, costituiscono il nostro CARATTERE.

Abbiamo detto che i bit possono assumere solo due valori, 0 o 1: un sistema numerico che è rappresentato solo da due simboli, si dice BINARIO.

Come si sa, il nostro sistema di numerazione abituale è quello DECIMALE: con i dieci simboli messi a disposizione possiamo rappresentare tutti i valori che vogliamo.

Ora, essendo anche il sistema binario un sistema di numerazione a tutti gli effetti, significa che anche con questo possiamo rappresentare tutti i valori che vogliamo ed eseguire tutte le comuni operazioni che facciamo con i decimali.

Senza entrare nel dettaglio della aritmetica binaria, quello che serve sapere ora è che una serie di simboli "0" e di "1" nel sistema binario costituiscono un normalissimo numero nel nostro sistema decimale.

Questo mi sugerisce sempre una considerazione: se noi, come uomini, avessimo solo due dita, il sistema di numerazione più logico sarebbe quello binario!

Se vi ricordate, il Byte è costituito da 8 Bit, che possono essere messi a 1 oppure a 0: l'insieme di questi 0 e 1 messi in un certo ordine costituisce un numero binario, che può essere trasformato in decimale con una semplice regola che i lettori abituali dovrebbero già sapere:

numerando gli 8 bit da 0 a 7, partendo da destra, e prendendo un byte a caso, otteniamo la seguente corrispondenza:

num. bit	7	6	5	4	3	2	1	0
valore	1	0	0	1	1	0	0	1

Per ottenere il valore decimale, occorre moltiplicare il valore di ogni bit ("0" oppure "1") per 2 elevato al numero di bit corrispondente, e sommare tutti i risultati così ottenuti; nel nosto caso sarà:

1*2↑7+
0*2↑6+
0*2↑5+
1*2↑4+
1*2↑3+
0*2↑2+
0*2↑1+
1*2↑0=

153

Ricordiamo che 2↑0=1: qualsiasi numero elevato alla potenza nulla fornisce come risultato 1 e non 0 come potrebbe sembrare a prima vista.

A questo punto, dire che un carattere è costituito da 8 byte equivale a dire che un carattere è formato da 8 numeri, e sono questi numeri ad essere memorizzati nella ROM: quando battiamo un tasto qualsiasi, il computer sa che deve andare nella ROM per prelevare quel particolare insieme di numeri che formeranno il simbolo premuto!

Il CBM 64, come il VIC 20 ed C 16, possiede due set di caratteri: i caratteri MAIUSCOLI e GRAFICI ed i caratteri MINUSCOLI e MAIUSCOLI (entrambi con i rispettivi reverse, ciò gli stessi caratteri in negativo).

Ogni set completo è formato da 256 caratteri: visto che ogni carattere è composto da 8 byte, un set completo occupa ben 256 8=2048 byte (2K di memoria: due set, 4K).

Per quanto riguarda il 64, questi byte partono dalla locazione 53248, e sono suddivisi come segue:

Indirizzo	Contenuto
53248	Car. Maiuscoli
53760	Car. Grafici
54272	Car. Maius. reverse
54784	Car. Graf. reverse
55296	Car. Minuscoli
55808	Car. Maius. e Graf.
56320	Car. Minus. rev.
56832	Car. Maius/graf/rev.

N.B. Nel set Minuscoli - Maiuscoli sono presenti comunque alcuni simboli grafici.

I due set, NON possono comparire contemporaneamente sullo schermo: per passare dall'uno all'altro occorre premere il tasto "Commodore" insieme col tasto Shift, oppure digitare POKE53272,21 per il passaggio al set Maiuscolo, mentre POKE53272,23 per il passaggio al Minuscolo.

Lo stesso passaggio può essere ottenuto con i CHR$(X): sfogliate la tabella del manuale e trovate voi i valori x da inserire tra le parentesi.

*** Una nota superflua: il 64, al momento dell'accensione, si trova nel set Maiuscolo - Grafico.***

Il primo carattere ad essere memorizzato nella Rom è il simbolo di chiocciolina ottenibile, tra l'altro, con PRINT CHR$(64): la Rom sarà, cioè, occupata dai numeri che formano tale carattere a partire dalla locazione 53248 fino a 53255.

Questo simbolo è anche il primo della tabella dei codici dello schermo: ciò significa che i singoli caratteri sono memorizzati nella Rom seguendo l'ordine di tale tabella (vedi manuale del computer).

Sempre in questa tabella troviamo le scritte serie 1 e serie 2, oppure set 1 e set 2; questo è riferito ai due set del computer: lo stesso codice (numero) può dare due caratteri diversi, a seconda del set in cui operiamo.

Vediamo ora come si fa a trovare la posizione di una qualsiasi lettera nella Rom.

Supponiamo di voler sapere dove si trovano i byte che determinano la Z, nel set Maiuscolo; dalla tabella dei codici, osserviamo che il codice della Z è 26.

Per ottenere la locazione del primo byte, occorre calcolare:
53248+8 26=53456

Tale byte, più i 7 successivi (fino a 53463), determinano la struttura- griglia del carattere "Z".

Il programma

Guardando le figure 1 e 2, capirete lo scopo del programma.

Si tratta di una dimostrazione di quanto detto finora: potrete vedere come sono costruite tutte le lettere e tutti i caratteri grafici di entrambi i set.

Sempre guardando le figure 1 e 2, abbiamo sulla sinistra la configurazione di tutti i BIT degli 8 BYTE costituenti la lettera A (set 1: figura 1 - set 2: figura 2), mentre, sulla destra, abbiamo i corrispondenti valori decimali.

Nella parte centrale è visualizzato il carattere, così come sarà riprodotto sullo schermo.

Dopo aver dato il RUN, verrà posta la domanda:
MAIUSCOLO (F1) O
MINUSCOLO (F2) ?.

Rispondendo col tasto F1 potrete analizzare tutti i caratteri del set Maiuscolo - Grafico, mentre rispondendo F2 (F1+shift), entrerete nel set Minuscolo.

La domanda successiva (CON COSA ?), riguarda semplicemente il simbolo con cui volete costruire il carattere (nelle fig. 1 e 2, è stato scelto l'asterisco "*").

Il carattere da analizzare, è prelevato dalla locazione dello schermo 1445 (linea 870): è necessario, perciò, copiare esattamente tutti i vari messaggi del listato (tranne le solite REM che potete anche evitare di trascrivere).

Alla fine della radiografia sul carattere, premendo Y esiste la possibilità di stampare su carta la radiografia stessa.

*** Attenzione: le POKE della subroutine Legge in Rom sono tra le più pericolose: un piccolo errore porta all'inchiodamento del computer!!!.

Consiglio perciò, come al solito, di registrare il programma prima di lanciarlo.

Sul significato di queste Poke torneremo in un altro articolo***

Giancarlo Castagna

```
BINARIO                                     DEC
0  0  0  1  1  0  0  0       **             24
0  0  1  1  1  1  0  0      ****            60
0  1  1  0  0  1  1  0     **  **           102
0  1  1  1  1  1  1  0     ******           126
0  1  1  0  0  1  1  0     **  **           102
0  1  1  0  0  1  1  0     **  **           102
0  1  1  0  0  1  1  0     **  **           102
0  0  0  0  0  0  0  0                      0
```

Fig. 1

```
BINARIO                                     DEC
0  0  0  0  0  0  0  0                      0
0  0  0  0  0  0  0  0                      0
0  0  1  1  1  1  0  0      ****            60
0  0  0  0  0  1  1  0         **           6
0  0  1  1  1  1  1  0      *****           62
0  1  1  0  0  1  1  0     **  **           102
0  0  1  1  1  1  1  0      *****           62
0  0  0  0  0  0  0  0                      0
```

Fig. 2

```
100 REM  COMMODORE 64
120 REM  CARATTERI AL
130 REM  MICROSCOPIO
140 :
150 REM  GIANCARLO CASTAGNA
190 :
200 PRINTCHR$(8)CHR$(147);
210 PRINTSPC(5)"*****************
    **********"
220 PRINTSPC(5)"* CARATTERI AL MIC
    ROSCOPIO *"
230 PRINTSPC(5)"*
                 *"
240 PRINTSPC(5)"*               @C
    AST 1985 *"
250 PRINTSPC(5)"*****************
    **********"
260 PRINT TAB(41)"MAIUSCOLO ('F1')
     O MINUSCOLO ('F2') ?"
270 GET R$
280 IF R$=CHR$(133) THEN PRINTCHR$
    (142):AD=53248:GOTO 310:REM  F
    1
290 IF R$=CHR$(137) THEN PRINTCHR$
    (14):AD=55296:GOTO 310:REM  F2
300 GOTO 270
310 PRINT TAB(85);:INPUT "CHE CARA
    TTERE ";S1$
320 S1=LEN(S1$):IF S1>1 THEN PRINT
    CHR$(145)CHR$(145)CHR$(145);:G
    OTO 310
330 PRINT TAB(45);:INPUT "CON COSA
    ";C$
340 C=LEN(C$):IF C>1 THEN PRINTCHR
    $(145)CHR$(145);:GOTO 330
350 GOSUB 860:PRINTCHR$(147)
360 PRINTCHR$(19)SPC(8)CHR$(18)" B
    INARIO "SPC(15)" DEC."CHR$(17)
370 REM   *******************
380 REM   * RIGHE CARATTERE *
390 REM   *******************
400 FOR I=0 TO 7:DE=CA(I)
410 REM   ************************
420 REM   * DECIMALE (DE) - BINARI
    O *
430 REM   ************************
440 FOR K=7 TO 0 STEP -1
450 QU=INT(DE/2):RE(K)=DE-2*QU
460 DE=QU:NEXT
470 REM   ********************
480 REM   * STAMPA CARATTERE *
490 REM   ********************
500 FOR T=0 TO 7:PRINTRE(T);:NEXT
510 FOR T=0 TO 7:IF RE(T)=0 THEN P
    RINTCHR$(32);:GOTO 530
520 PRINTC$;
530 NEXT:PRINTCA(I)
540 NEXT
550 REM   *********************
560 REM   * STAMPA SU CARTA ? *
570 REM   *********************
580 PRINT TAB(208)"VUOI STAMPARE (
    Y/N) ?"
590 GET R$:IF R$="Y" THEN 720
600 IF R$="N" THEN 650
610 GOTO 590
620 REM   **************
630 REM   * RICOMINCIA *
640 REM   **************
650 PRINT TAB(160) TAB(127)CHR$(18
    )" F7 - CONT.  F5 - FINE "
660 GET R$:IF R$=CHR$(136) THEN 20
    0
670 IF R$=CHR$(135) THEN END
680 GOTO 660
690 REM   ***********
700 REM   * MPS 803 *
710 REM   ***********
720 OPEN 4,4
730 PRINT#4,CHR$(18);CHR$(16);"07"
    ;" BINARIO ";CHR$(16);"32";" D
    EC ";CHR$(148)
740 LO=0:A$=""
750 FOR I=1064+LO TO 1103+LO
760 PP=PEEK(I):IF PP<32 THEN PP=PP
    +64:GOTO 790
770 IF PP>63 AND PP<96 THEN PP=PP+
    32:GOTO 790
780 IF PP>96 THEN PP=PP+64
790 A$=A$+CHR$(PP):NEXT
800 PRINT#4,A$:LO=LO+40:IF LO<400
    THEN A$="":GOTO 750
810 FOR I=1 TO 4:PRINT#4:NEXT:CLOS
    E 4
820 GOTO 650
830 REM   ****************
840 REM   * LEGGE IN ROM *
850 REM   ****************
860 POKE 56334,PEEK(56334) AND 254
    :POKE 1,PEEK(1) AND 251
870 PE=PEEK(1445)
880 FOR I=0 TO 7:CA(I)=PEEK(AD+I+P
    E*8):NEXT
890 POKE 1,PEEK(1) OR 4:POKE 56334
    ,PEEK(56334) OR 1
900 RETURN
```

Simula robot

Una nuova febbre si sta diffondendo in questi ultimi mesi tra gli appassionati di informatica ed elettronica: quella del personal robot.

Chi, tra coloro che hanno visto i servizi dedicati all'EXPO tenutasi recentemente a Tsukuba, in Giappone, non avrebbe fatto un patto con il diavolo pur di avere il robot che suona il pianoforte o quello che cammina come un essere umano, con due sole gambe?!?

Purtroppo per ora tali fantastiche macchine non sono che un sogno per noi poveri hobbisti.

Non scoraggiamoci, però: già da qualche tempo è in circolazione un braccio meccanico che si può interfacciare ai personal computer e di cui si possono programmare i movimenti.

Vi sono poi altri piccoli automi "semoventi" (apparsi anche in varie trasmissioni televisive) in continua evoluzione.

Nell'attesa che tali robot diventino più intelligenti (e che i prezzi calino), possiamo fin d'ora prendere confidenza con i loro comportamenti simulandoli con il nostro Commodore 64.

Il programma qui riportato si propone di simulare due delle fondamentali "performance" dei robot: il movimento di un braccio e la presa di oggetti.

Uno studio dei problemi che si presentano nella progettazione dei robot risolto, in questo articolo, con una simulazione semplice ed interessante.

Introduzione

Muovere un braccio verso un oggetto ed afferrarlo è per noi umani una cosa semplicissima, che non richiede particolare concentrazione e impegno: lo facciamo da sempre.

Ma proviamo a pensare come avviene l'operazione.

1/ Vediamo l'oggetto e ne calcoliamo la posizione rispetto a noi.
2/ Ruotiamo braccio ed avambraccio nella direzione dell'oggetto.
3/ Controlliamo visivamente se abbiamo raggiunto l'oggetto.
4/ Se ciò non è avvenuto torniamo al punto 2.
5/ Afferriamo l'oggetto.

Le fasi 2 e 3 si svolgono ad una tale velocità che il movimento risulta continuo ed unico ma provate a ripetere la stessa operazione chiudendo gli occhi nella fase 2.

Vi accorgerete che difficilmente riuscirete a raggiungere con precisione la posizione desiderata al primo tentativo.

Per un robot (supposto che disponga di una telecamera o di un radar) operare in questo modo non sarebbe molto conveniente, a meno che non riesca a riprodurre una delle doti fondamentali dell'uomo: l'esperienza, che porta a individuare immediatamente i movimenti appropriati tra gli infiniti possibili.

Esiste però un metodo che può assicurare al nostro robot (sempre se dotato di "vista") una veloce e sicura riuscita nell'impresa (per non complicare troppo il discorso farò riferimento ad uno spazio a due sole dimensioni).

La teoria

Immaginiamo di avere, come in figura 1, un braccio composto da due parti: R1 e R2.

L'estremo libero di R1 è "piantato" nella posizione C1 e vogliaao fare raggiungere all'estremo libero di R2 (cioè la mano) la posizione C2.

Immaginiamo ora di spezzare il braccio all'altezza del gomito, e di piantare l'estremità corrispondente alla mano (R2) nel punto C2.

L'impostazione del problema è ora cambiata dato che dobbiamo congiungere R1 e R2 allo scopo di ottenere il collegamento C1-C2.

Ebbene noi sappiamo che R1 e R2, ruotando intorno ai loro rispettivi fulcri C1 e C2, descrivono due circonferenze e i

Figura 1: Intersezione di circonferenze

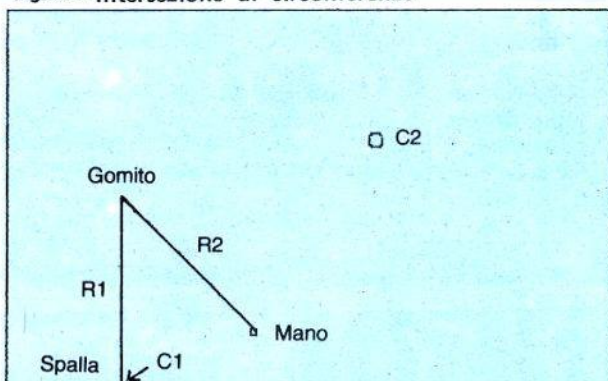



Figura 2

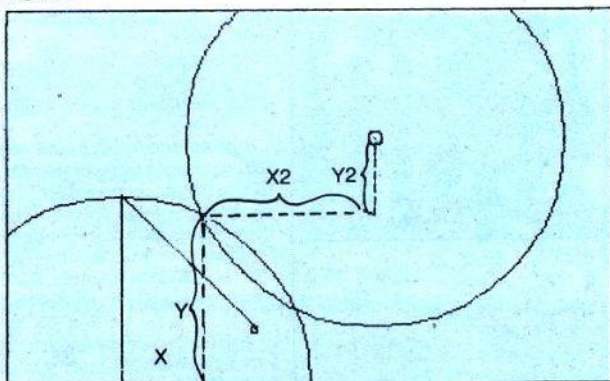
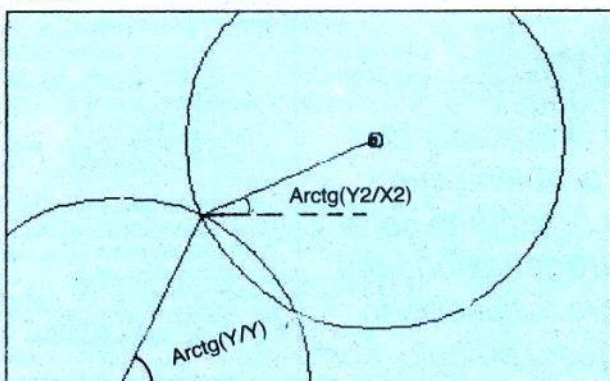



Figura 3

Conoscendo la posizione di un oggetto (C2), rispetto alla "spalla" (C1), e conoscendo la lunghezza di braccio (R1) ed avanbraccio (R2) è possibile, tramite la risoluzione di un sistema tra le due circonferenze di centro C1 e C2 e di raggio R1 ed R2 ripettivamente, trovare le due posizioni possibili (i punti d'intersezione delle circonferenze) del gomito che consentono di portare la "mano" nel punto desiderato.

Conoscendo ciò è facile determinare gli angoli di rotazione corrispondenti.

Teo Rusconi ha appena sfatato la leggenda secondo la quale i floppy disc sono tutti uguali

Difatti sembrano tutti uguali finchè non si osserva con attenzione il jacket. Qui termina l'uguaglianza.

La maggior parte delle società costruttrici sigillano i dischi un punto qui, un punto là, lasciando parte dei lembi non sigillati.

Prima o poi ai lembi accadono cose naturalissime: si gonfiano, si curvano, si raggrinziscono... in poche parole si aprono.

Con penne, matite, unghie persino un ragazzino di quattro anni come Teo può infilarsi in quegli spazi aperti.

Naturalmente è un danno enorme perchè se si inserisce qualcosa di molle e slabbrato nel disc-drive quest'ultimo può incepparsi; si può rovinare la testina e si possono perdere i dati. Questo può accadere con gli abituali sistemi di chiusura ma non con i dischetti Memorex che usa un procedimento esclusivo chiamato "Solid-Seam Bonding".

Con questo sistema ogni singolo millimetro quadrato dei lembi di tutti i dischi Memorex viene sigillato ermeticamente, rendendoli più rigidi e più resistenti.

È un sistema che consente al floppy disc di sostenere ogni assalto, che impedisce alla testina di rovinarsi e ai dati di andare perduti.

Il che sta a dimostrare che un floppy disc Memorex non è uguale a tutti gli altri: è migliore. E il sistema di saldatura è solo un esempio della cura infinita con cui viene prodotto ogni floppy disc Memorex; sia esso da 8", da 5 1/4" o il nuovo 3 1/2". Questa estrema accuratezza dà la garanzia che ogni disco Memorex è al 100% perfetto.

La prossima volta che acquistate un floppy disc - o qualche centinaio - ricordate: non tutti i dischetti sono uguali...
Memorex vi mette al riparo da qualsiasi inconveniente.

MEMOREX MEMOREX MEMOREX

è importante scegli
MEMOREX
A Burroughs Company

BURROUGHS-MEMOREX S.P.A.
Divisione Computer Media
Via Ciro Menotti, 14 Tel. 02/718551
20129 MILANO MI

punti in cui tali circonferenze si intersecano indicano le possibili congiunzioni dei due bracci, cioè danno la soluzione del problema iniziale.

Conoscendo la posizione C2 da raggiungere (rispetto alla "spalla" C1) e le lunghezze di braccio (R1) ed avambraccio (R2), si possono calcolare i punti d'intersezione (se esistono) tra la circonferenza di centro C1 e raggio R1 e quella di centro C2 e raggio R2.

In seguito basta calcolare gli angoli al centro corrispondenti e fare ruotare i semibracci di conseguenza.

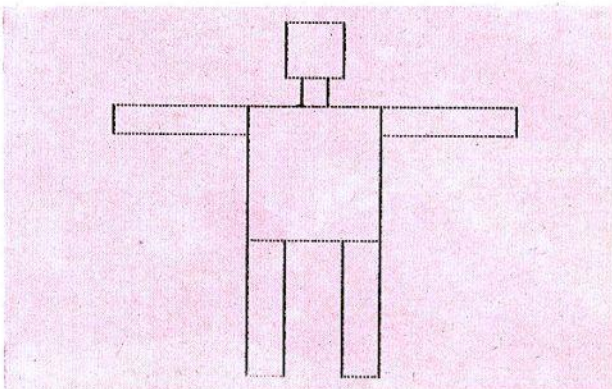
Nelle figure 1-2-3 viene illustrato l'intero processo di calcolo.

Questo sistema presenta però un inconveniente: vale solamente per bracci composti di due sole parti.

Consideriamo comunque che tale composizione degli arti sembra essere la regola prevalente in natura.

Per meglio comprendere tutto quanto detto, vi propongo un programma che permette di muovere un braccio con entrambi i metodi.

Rotazione semibraccio

La rotazione avviene intorno al fulcro (F). L'angolo formato dai due semibracci rimane immutato.

Umanoide

I tramite il braccio - robot dovete accostare gli sprite presenti sullo schermo in modo da formare questa figura.

Il programma simulatore

Perchè tale listato giri occorre prima avere caricato le routine grafiche pubblicate sul N. 14 di C.C.C. (e riproposte sulla cassetta Commodore Club N. 4). Fatto ciò, potete caricare il programma pubblicato in queste pagine e dare il RUN.

Vi verrà chiesto di quante parti (semibracci) volete sia formato il vostro braccio meccanico.

Potete rispondere con un numero compreso tra 1 e 9.

Premuto il tasto RETURN vi verrà domandato di fornire la misura (in punti-schermo) di ogni semibraccio.

Una volta terminata la fase di input lo schermo passa in alta risoluzione e, dopo un paio di secondi, vengono visualizzati il braccio da voi scelto, terminante con una mano rudimentale, e altri sei oggetti (sprite).

Per manovrare il braccio potete utilizzare i tasti 1, 2, 3, ... 9 che fanno ruotare il semibraccio corrispondente di 10 gradi intorno al proprio fulcro.

La rotazione avverrà in senso orario. Premendo contemporaneamente il tasto SHIFT la rotazione sarà invece antioraria.

Come è logico i semibracci "a valle" di quello che ruota si spostano solidalmente con questo (vedere figura 4).

Premendo il tasto "-" (meno) l'angolo di rotazione diventa di 1 grado. Premendo "+" torna a 10 gradi.

I tasti "C" e "A" servono rispettivamente a chiudere e ad aprire la mano.

Se un oggetto si trova nella mano quando chiudete viene "afferrato" e tenuto finchè non riaprite.

Veniamo ora agli oggetti-sprite: partendo da sinistra, trovate le gambe, le braccia, la testa e il tronco di un "umanoide" (fig.5), che dovete cercare di assemblare servendovi del braccio-robot.

Per fare ciò, dovete ruotare i semibracci intorno alle "articolazioni" fino a raggiungere gli oggetti con la mano, dopo di che afferrarli e accostarli in modo da ottenere l'umanoide della figura.
Lo scopo di tutte queste operazioni è di farvi rendere conto come sia complicato portare in modo empirico la mano nel punto desiderato.

Più semplice è utilizzare il metodo matematico precedentemente descritto.

Come già detto, questo funziona solo con bracci formati da due parti, quindi, quando vi viene chiesto "numero di semibracci", rispondete 2.

Vi accorgerete che sullo schermo, oltre al braccio e ai sei sprite già citati, comparirà una crocetta nera: il vostro mirino.

Potete muoverlo tramite i tasti del cursore.

Se premete il tasto "-" lo spostamento sarà di un pixel per volta mentre premendo il tasto "+" la crocetta si muove di dieci pixel.

Il centro del mirino indica il punto che volete raggiungere con la mano.

Quando avrete portato la crocetta nel punto desiderato, premete il tasto "M" e

il braccio si muoverà e raggiungerà la posizione segnalata.

Se tale posizione è irraggiungibile (cioè se non vi è intersezione tra le circonferenze) il mirino lampeggerà e non vi sarà spostamento.

Per aprire e chiudere la mano valgono sempre i tasti "A" e "C"; inoltre potete anche in questo caso muovere il braccio "manualmente" utilizzando i tasti "1" e "2".

Il listato

Se non avete voglia di digitare il programma tutto d'un fiato, potete fermarvi alla linea 1510.

Funzionerà ugualmente in modo "manuale" poichè le linee dalla 1540 in poi riguardano esclusivamente l'implementazione del metodo delle circonferenze.

Naturalmente dovrete stare attenti a non toccare i tasti del cursore e il tasto "M" mentre il programma è in esecuzione, altrimenti si genererà un "UNDEF'D STATEMENT ERROR" (alternativamente potete cancellare le linee dalla 900 alla 940 e ogni rischio verrà eliminato).

Nel listato ho inserito molte spiegazioni per rendere più facile la comprensione della logica dello stesso.

Potrete così modificare, se ne avrete voglia, tale programma e magari trasformarlo in un giochino.

Danilo Toma

```
100 REM  SIMULA-ROBOT
110 :
130 REM  PROGRAMMA PER
140 REM  COMMODORE 64
150 REM  (OCCORRONO LE ROUTINE
160 REM  GRAFICHE PUBBLICATE
170 REM  SUL N. 14 DI C.C.C.)
180 REM
190 REM  DI DANILO TOMA
220 :
230 REM ********************************
240 REM ****      INIZIALIZZA       ****
250 REM ********************************
260 POKE650,128:POKE53280,11:←TEXT6,3
270 S=28:E=10:T=-1
280 X(1)=-150:Y(1)=-100
290 :
300 INPUT"██NUMERO DI SEMIBRACCI (1-9)";N:IFN<1ORN>9THEN300
310 FORI=1TON
320 PRINT"█LUNGHEZZA SEMIBRACCIO";I;
330 INPUTL(I):IFL(I)<0THENPRINT"□□□":GOTO320
340 R(I)=.5*π-π*(I-1)/N
350 NEXT
360 :
370 ←CLEAR:←MGRAF6,1,0,10
380 V=53248
390 :
400 REM ****   MIRINO    ****
410 FORI=51968TO52030STEP3
420 POKEI,0:POKEI+1,24:POKEI+2,0:NEXT
430 FORI=51998TO52000:POKEI,255:NEXT
440 POKEV-8,44:POKEV+39,0
450 POKEV,172:POKEV+1,159
460 :
470 REM ****   CORPO    ****
480 FORI=51712TO51774:POKEI,255:NEXT
490 POKEV-2,40
500 :
510 REM ****   GAMBE    ****
520 FORI=51776TO51838STEP3
530 POKEI,0:POKEI+1,126:POKEI+2,0:NEXT
540 POKEV-7,41:POKEV-6,41
550 :
```

```
560 REM ****   BRACCIA   ****
570 FORI=51840TO51851:POKEI,255:NEXT
580 FORI=51852TO51902:POKEI,0:NEXT
590 POKEV-5,42:POKEV-4,42
600 :
610 REM ****   TESTA   ****
620 FORI=51904TO51933STEP3
630 POKEI,7:POKEI+1,255:POKEI+2,224:NEXT
640 FORI=51934TO51949STEP3
650 POKEI,0:POKEI+1,255:POKEI+2,0:NEXT
660 FORI=51950TO51966:POKEI,0:NEXT
670 POKEV-3,43
680 :
690 REM ****   POSIZIONA SPRITES   ****
700 FORI=1TO6
710 POKEV+I*2,I*35+45:POKEV+I*2+1,230
720 POKEV+I+39,5
730 NEXT
740 POKEV+21,127+(N<>2):POKEV+27,126
750 :
760 A=1:GOSUB1060
770 :
780 REM ********************************
790 REM * ATTESA ED ESECUZIONE COMANDI *
800 REM ********************************
810 FORJ=0TO8:GETA$:NEXT
820 P=0:GETA$:IFA$=""THEN820
830 IFA$="A"THENC=0:T=-1:GOSUB1410:S=28:C=2:GOSUB1410
840 IFA$="C"THENC=0:GOSUB1410:S=21:C=2:GOSUB1350
850 IFA$="+"THENE=10
860 IFA$="-"THENE=1
870 IFA$>"0"ANDA$<":"THENP=-E*π/180
880 IFA$>" "ANDA$<"*"THENP=E*π/180:A$=CHR$(ASC(A$)+16)
890 IFN<>2THEN960
900 IFA$="□"THENPY=-E:PX=0:GOSUB1540
910 IFA$="■"THENPY=E:PX=0:GOSUB1540
920 IFA$="■"THENPX=E:PY=0:GOSUB1540
930 IFA$="■"THENPX=-E:PY=0:GOSUB1540
940 IFA$="M"THENGOSUB1670
950 :
960 IFP=0THEN810
970 A=VAL(A$):IFA>NTHEN810
980 GOSUB1060
990 GOTO810
1000 :
1010 REM ******************************
1020 REM ****     SUBROUTINES     ****
1030 REM ******************************
1040 :
1050 REM ** CALCOLA NUOVE POSIZIONI **
1060 TX(A)=X(A):TY(A)=Y(A)
1070 FORI=ATON
1080 R(I)=R(I)+P
1090 TX(I+1)=X(I+1)
```

```
1100 X(I+1)=X(I)+COS(R(I))*L(I)
1110 TY(I+1)=Y(I+1)
1120 Y(I+1)=Y(I)+SIN(R(I))*L(I)
1130 NEXT
1140 :
1150 REM ** EVIDENZIA LO SPOSTAMENTO **
1160 ←COLOR3:FORI=ATON
1170 ←DRAWX(I),Y(I),0,X(I+1),Y(I+1),0
1180 NEXT
1190 ←ARCX(N+1),Y(N+1)-10,0,S,14,-π/4,1.25*π,π/2
1200 :
1210 REM * CANCELLA VECCHIA POSIZIONE *
1220 ←COLOR0:FORI=ATON
1230 ←DRAWTX(I),TY(I),0,TX(I+1),TY(I+1),0:NEXT
1240 ←ARCTX(N+1),TY(N+1)-10,0,S,14,-π/4,1.25*π,π/2
1250 :
1260 REM * PASSA ALLA NUOVA POSIZIONE *
1270 X=X(N+1)+172:Y=159-Y(N+1):GOSUB1440
1280 ←COLOR1:FORI=ATON
1290 ←DRAWX(I),Y(I),0,X(I+1),Y(I+1),0:NEXT
1300 ←COLOR2:←ARCX(N+1),Y(N+1)-10,0,S,14,-π/4,1.25*π,π/2
1310 RETURN
1320 :
1330 REM ** VERIFICA SE HA AFFERRATO UNO SPRITE **
1340 :
1350 FORJ=2TO12STEP2
1360 SX=PEEK(V+J)-256*(0<>(PEEK(V+16)AND2↑(J/2)))
1370 IFX<SX-8ORX>SX+8THEN1400
1380 IFY<PEEK(V+J+1)-8ORY>PEEK(V+J+1)+8THEN1400
1390 T=J:J=16
1400 NEXT
1410 ←COLORC:←ARCX(N+1),Y(N+1)-10,0,S,14,-π/4,1.25*π,π/2
1420 :
1430 REM ** SE T <> -1 HA IN MANO UNO SPRITE **
1440 IFT=-1THEN RETURN
1450 REM ***  SPOSTA LO SPRITE  ***
1460 X=INT(X)
1470 IFX<0ORX>344ORY<0ORY>255THENPOKEV+21,PEEK(V+21)AND(255-2↑(T/2)):RETURN
1480 IFX<256THENPOKEV+16,PEEK(V+16)AND(255-2↑(T/2))
1490 IFX>255THENPOKEV+16,PEEK(V+16)OR2↑(T/2):X=X-256
1500 POKEV+T,X:POKEV+T+1,Y:POKEV+21,PEEK(V+21)OR2↑(T/2)
1510 RETURN
1520 :
1530 REM ***  SPOSTA IL MIRINO  ***
1540 XX=X:YY=Y
1550 X=PEEK(V)+PX+256*(PEEK(V+16)AND1)
1560 Y=PEEK(V+1)+PY
1570 TT=T:T=0
1580 GOSUB1470
1590 T=TT:X=XX:Y=YY
1600 RETURN
1610 :
1620 REM ***  CALCOLA SPOSTAM. NECESSARIO  ***
1630 REM ***  PER RAGGIUNGERE IL MIRINO    ***
```

```
1640 :
1650 REM ** CERCA INTERSEZIONE TRA LE CIRCONFERENZE **
1660 REM ** DI CENTRO [XX,YY] E [X(1),Y(1)] **
1670 XX=-171+PEEK(V)+256*(PEEK(V+16)AND1)
1680 YY=155-PEEK(V+1)
1690 A1=-2*X(1):B1=-2*Y(1):C1=X(1)*X(1)+Y(1)*Y(1)-L(1)*L(1)
1700 A2=-2*XX:B2=-2*YY:C2=XX*XX+YY*YY-L(2)*L(2)
1710 A3=A1-A2:B3=B2-B1:C3=C2-C1
1720 A4=A3*A3+B3*B3
1730 B4=B3*(2*C3+A3*A1)+A3*A3*B1
1740 C4=C3*(C3+A3*A1)+A3*A3*C1
1750 DY=B4*B4-4*A4*C4
1760 IFDY<0THEN2110
1770 Y1=(-B4+SQR(DY))/(2*A4)
1780 C5=B3*Y1+C3
1790 X1=C5/A3
1800 :
1810 REM ** CALCOLA LE ROTAZIONI DA EFFETTUARE **
1820 DX(1)=X1-X(1):DY(1)=Y1-Y(1)
1830 DX(2)=XX-X1:DY(2)=YY-Y1
1840 TX(1)=X(1):TY(1)=Y(1)
1850 FORI=1TO2
1860 IFDX(I)=0ANDDY(I)>0THENAN(I)=π/2:GOTO1910
1870 IFDX(I)=0ANDDY(I)<0THENAN(I)=-π/2:GOTO1910
1880 AN(I)=ATN(DY(I)/DX(I))
1890 IFDX(I)<0ANDDY(I)<0THENAN(I)=AN(I)-π
1900 IFDX(I)<0ANDDY(I)>0THENAN(I)=AN(I)+π
1910 DR(I)=AN(I)-R(I)
1920 NEXT
1930 :
1940 REM ****  SUDDIVIDE LO SPOSTAMENTO  ****
1950 REM ****  IN 1, 2 OPPURE 3 PASSI   ****
1960 A=1:H=3
1970 IFABS(DR(1))+ABS(DR(2))<.8THENH=2
1980 IFABS(DR(1))+ABS(DR(2))<.4THENH=1
1990 FORJ=1TOH
2000 FORI=1TO2
2010 R(I)=R(I)+DR(I)/H
2020 TX(I+1)=X(I+1)
2030 X(I+1)=X(I)+COS(R(I))*L(I)
2040 TY(I+1)=Y(I+1)
2050 Y(I+1)=Y(I)+SIN(R(I))*L(I)
2060 NEXT
2070 GOSUB1160
2080 NEXT:RETURN
2090 :
2100 REM **** MIRINO LAMPEGGIANTE ****
2110 FORI=0TO3
2120 POKEV+39,1
2130 FORJ=0TO50:NEXT
2140 POKEV+39,0
2150 FORJ=0TO50:NEXT
2160 NEXT
2170 RETURN
```

UNA CASSAFORTE CHIAMATA COMMODORE 64

di **Fabio Sorgato**

Come rendere incopiabili i vostri programmi.

Sin da quando vennero messi in commercio i computer, le software-house decisero di difendere i propri prodotti evitando la pirateria (copia abusiva di programmi) con tecniche di protezione sempre più sofisticate. Vediamo qualche trucco per proteggere anche i vostri programmi personali.

I programmi che si proteggono nel modo che stiamo per descrivere generano inconvenienti anche ricorrendo alle stampanti, quindi non ci è possibile... stamparli. E' pertanto necessario seguire alla lettera le operazioni da compiere.

Supponiamo che il programma da proteggere sia:

100 PRINT"PIPPO":REM

Subito dopo la REM:

1/ scrivete due volte i doppi apici (SHIFT + 2);

2/ cancellatene uno premendo il tasto DELETE (il primo in alto a destra);

3/ premete 21 volte il tasto SHIFT insieme con DELETE (sembrerà che non accada nulla, ma fatelo lo stesso);

4/ premere un egual numero di volte il solo tasto DELETE (appariranno altrettanti caratteri "T" in reverse, cioè nero su bianco);

5/ digitate la parola PROTETTO.

Terminato di digitare, fate un List e se non avete fatto alcun errore vedrete in sostituzione della riga 100 la sola scritta PROTETTO.

Cos'è successo? Ogni T "reversata" ha cancellato, durante la fase di List, il carattere che la precedeva (tra cui il numero di riga Basic); così abbiamo ottenuto la visualizzazione della sola scritta.

Se invece della parola PROTETTO inserite spazi bianchi (terminanti col carattere di virgolette), renderete invisibile la riga Basic. Dato il RUN osserverete che la riga, sebbene non più visibile, funziona ancora.

L'unico problema è che per rendere invisibile un intero listato sarebbe necessario fare scomparire tutte le righe, con un lavoro lungo e noioso ed un notevole spreco di memoria.

```
100 PRINT"PIPPO":RE
    M█████████████████████
    ██████████PROTETTO
```

```
LIST

PROTETTO
```

```
100 PRINT"PIPPO":RE
    M L
```

```
LIST

?SYNTAX  ERROR
```

Cancellate il programma (NEW) e scrivete:
100 PRINT"PIPPO":REM L

Il carattere dopo il REM non è una L maiuscola, ma il carattere ottenuto con SHIFT + "L". Provate a fare un List. Otterrete ?SYNTAX ERROR, con conseguente blocco del listato.

Combinando le due tecniche apprese, sarà possibile rendere più difficile la lettura del listato.

Per mettere in grave difficoltà anche le persone più esperte sarebbe consigliabile terminare ogni riga con un REM seguito da L+SHIFT. Infatti comprendere il programma sarà un lavoro lungo e noiosissimo che scoraggerà la prosecuzione della sprotezione.

Un problema

Purtroppo, nei casi visti, il programma potrà comunque essere riprodotto semplicemente con un'operazione di SAVE senza curarsi di come è fatto "dentro".

Affrontiamo quindi questo problema. Innanzitutto è necessario sapere che quando un programma viene caricato da cassetta, il suo nome viene conservato in una precisa zona di memoria, (da 833 a 1023, ben più lunga dei sedici caratteri ufficialmente consentiti),che non verrà cancellato fino alla successiva operazione che chiama in causa il registratore e che, ovviamente, verrà perso in caso di spegnimento del computer, reset, SYS64738, eccetera.

Proviamo a registrare il solito programma (100 PRINT"PIPPO") così:

SAVE"PROGRAMMA"

Ricarichiamolo e, ad operazione avvenuta, digitiamo direttamente da tastiera i seguenti comandi:

CLR:FORT=833TO1023:A$=A$
+CHR$(PEEK(T)):NEXT:PRINTA$

Solo ora la variabile A$ conterrà il vero nome del programma appena caricato che, nella gran parte dei casi, ha valore solo nei primi 16 caratteri.

Tutto ciò ci permetterà di mettere a punto una tecnica di protezione.

Il trucco è semplice: si nasconde nel nome del programma un carattere (che assumerà la funzione di "controllo") e mediante una riga nascosta o un listato non listabile si controlla se il carattere previsto è presente. In caso contrario si predispongono risultati imprevedibili. L'accanito copiatore, ignaro del carattere di controllo registrerà, il programma solo con il nome visibile o magari con un altro nome e cadrà nella trappola.

Facciamo una prova con il listato 1: nelle righe 10 e 20 abbiamo il solito controllo nascosto.

Senza dare RUN registriamo il programma con:

A$="?"+CHR$(20)+

"PROGRAMMA":SAVEA$

In A$ il "?" è il carattere di controllo (vedi riga 20), CHR$(20) è una T reversata che rende invisibile il "?" cancellandolo.

Finita la registrazione ricarichiamo il programma, diamo RUN e, se tutto è stato fatto nel modo esatto, funzionerà.

Ora diamo:
SAVE"PROGRAMMA"
cioè il nome che è quello visto al FOUND durante il caricamento.

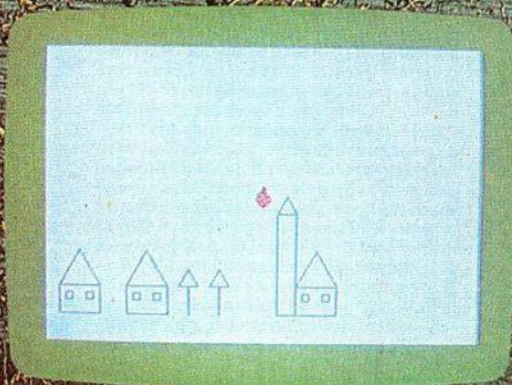
Ricaricatolo, diamo RUN e vedremo il computer tornare allo stato in cui si trovava al momento dell'accensione, con la conseguente cancellazione del programma: la protezione ha funzionato bene!

I pirati più astuti potranno però sempre copiare il programma con un duplicatore di cassette musicali. In questo caso, purtroppo, non c'è assolutamente nulla da fare!

A proposito, non mandateci listati da pubblicare protetti: fidatevi almeno di noi!

Imparando ad imparare

di M.L.Nitti e D.Matturro

Il paese
elettronico
Come costruire un
paese elettronico
con un quadrato,
un triangolo e
qualche linea.

Le argomentazioni prese in esame nella scorsa puntata di "Imparando ad imparare" (C.C.C. Ottobre 85) portavano alla soluzione "elettronica" di piccoli problemi secondo procedimenti "lineari". Procedimento lineare significa: una serie di istruzioni che si susseguono una dopo l'altra verso la fine del programma senza essere rieseguite.

Tali procedimenti, in realtà, vengono usati molto poco poichè risolvono i problemi con grande dispendio di lavoro (bisogna scrivere tanto!!!) e di memoria. Sulla memoria vale la pena aprire una piccola parentesi, dedicate a coloro che partono da zero.

Il "64" ha perso la memoria?

Tutti i computer hanno una memoria (di solito....)! Tutte le memorie sono più o meno grandi, più o meno piccole; questo significa che se le memorie sono misurabili, deve necessariamenete esistere un'unità di misura. Sia chiaro, nessuno dirà mai di possedere un computer con "tot" centimetri o "tot" chili di memoria, infatti l'unità di misura in questione si chiama Byte e non si riferisce nè alla lunghezza nè al peso, bensì alla quantità di informazioni contenibili dalla memoria stessa.

Spesso si sente dire che un Byte corrisponde ad un carattere. E' vero ma fino ad un certo punto; più propriamente un Byte è un contenitore di cifre binarie. Infatti un byte è composto di otto bit e un bit corrisponde ad una cifra binaria. Le cifre binarie si rapresentano con i numeri 0 e 1; se scriviamo 001, indicheremo un numero binario composto di tre cifre o meglio di tre bit. Il numero binario 11111111 corrisponde al numero decimale 256, è composto di otto bit e rappresenta la grandezza numerica massima contenibile in un byte.

Anche se un byte può incamerare un numero di tre cifre, in verità, dal puntò di vista alfabetico, il byte non può contenere più di un carattere perchè ogni carattere viene scritto nella memoria sotto forma di codice numerico (es. la lettera "a" corrisponde al numero 65); se mettessimo più di un codice numerico nello stesso byte non saremmo più in grado di riconoscere i codici originali in un secondo momento.

A questo punto qualcuno dirà: "ma allora il mio "64" ha perso la memoria!". Effettivamente, all'apparire della scritta "64K RAM System 38911 BASIC Bytes free" qualcuno potrebbe pensare di avere un Commodore con soli 38911 byte di memoria, ma non è così; il vecchio Commodore ha sempre la sua giusta "dose" di byte, questi però non sono tutti a disposizione dell'utente. La scritta dice, infatti, 38911 byte liberi per il Basic, gli altri sono occupati da programmi "speciali" che gestiscono la macchina. Altra lecita domanda: "Da dove arrivano? Io non li ho mai caricati in memoria!!". La scritta dice 64K RAM e sta a significare che sono disponibili 64000 bytes di memoria RAM, ma cosa vuol dire?. Significa che esistono vari tipi di memoria.

Senza entrare nei particolari, è utile sapere che la memoria RAM può essere sia scritta che letta e si cancella se spegnamo il computer; quindi quei programmi "intrusi" di prima non possono essere qui; infatti sono contenuti da un'altro tipo di memoria detta ROM e con particolari accorgimenti, quando si accende il computer occupano alcune zone della RAM togliendo una parte dei 64K bytes. Chiaramente la ROM non si cancella mai, però non può essere scritta.

Di questi particolari programmi ci riserviamo di parlare nel prossimo numero.

Come fare impazzire la "tartaruga"

Tornando al discorso iniziale, i procedimenti lineari, quindi, non vengono molto usati, perchè occupano tanta RAM; quasi sempre invece i programmi vengono costruiti mediante procedimenti ciclici.

Un ciclo (gli addetti ai lavori lo chiamano "loop") è una struttura ripetitiva; nella sua forma più elementare consiste in un gruppo di istruzioni rieseguite all'infinito. Attenzione! "All'infinito" non significa che non si possono fermare, si intende semplicemente che il programma continua a "girare" fino a quando non si preme il tasto RUN/STOP (vedi figura 1).

to. Chiaramente per fare questo occorre un istruzione particolare, un istruzione che gli amanti delle definizioni teoriche chiamano "salto incondizionato", e che in E.LI.ANA. si chiama LINEA.

Il comando LINEA fa si che l'esecuzione del programma "salti" ad un altra istruzione; l'istruzione LINEA è seguita da un parametro, che deve corrispondere al numero di riga in cui deve "saltare" l'esecuzione. Una volta effettuato il "salto", il programma viene eseguito da quel punto continuando in ordine crescente di linea in linea. Se cancelliamo le righe dalla 220 alla 270 del programma che esegue il quadrato e sostituiamo la riga 280 con:

```
280 LINEA 200
```

potremo ottenere una semplicissima ma esemplare procedura ciclica infinita. Infatti, digitando il comando ESEGUI potremmo vedere che la tartaruga disegna il quadrato voluto con una singolare variante: là dove prima interrompeva l'esecuzione ora continua, rieseguendo lo stesso quadrato senza soluzione di continuità (sembra impazzita!). L'arresto deve essere determinato "dall'esterno" premendo RUN/STOP.

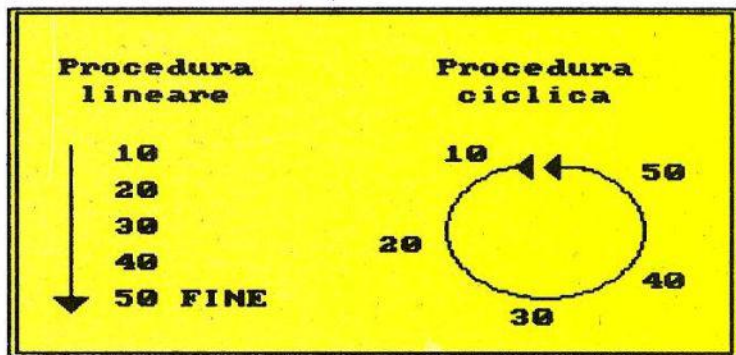



fig.1

Il programma "quadrato" visto nel numero precedente e riportato qui di seguito, può essere utilizzato per comprendere il concetto di struttura ciclica infinita.

```
100 ALBUM
110 MATITA
200 DESTRA 90
210 AVANTI 20
220 DESTRA 90
230 AVANTI 20
240 DESTRA 90
250 AVANTI 20
260 DESTRA 90
270 AVANTI 20
280 FINE
```

Le righe 220-270 non sono altro che la ripetizione della 200 e 210 (meno male che il quadrato ha solo quattro lati!) e potrebbero essere rieseguite per ottenere lo stesso effetto. Chiaramente per fare questo occorre un istruzione particolare, ...

Cosa c'è sotto il programma?

Anche il caso di cui sopra non trova grande riscontro nelle pratiche "programmatorie" dei più; ci sono altri sistemi per creare strutture cicliche e fare in modo che vengano eseguite un numero finito di volte, oppure che vengano eseguite fino al verificarsi di determinate condizioni in modo che non si debba intervenire dall'esterno.

Tralasciando per ora l'utilità dell'istruzione LINEA, sulla quale torneremo in seguito, introduciamoci in un nuovo modo di scrivere programmi che offre maggiori possibilità risolutorie.

Osservate questa nuova versione del programma "quadrato":

```
100 ALBUM
110 MATITA
120 SOTTOPROGRAMMA 200;4
130 FINE

200 DESTRA 90
210 AVANTI 20
220 TORNA
```

Le righe 200-220 costituiscono un sottoprogramma, e terminano con l'istruzione TORNA; per gli affezionati del BASIC quest'ultima corrisponde al RETURN. La riga:

```
120 SOTTOPROGRAMMA 200;4
```

costituisce la "chiamata" del sottoprogramma e contiene due parametri: numero di linea dove inizia il sottoprogramma da "chiamare" e numero di volte che questo deve essere eseguito. I conoscitori del BASIC avranno già capito che l'istruzione SOTTOPROGRAMMA corrisponde al GOSUB con l'aggiunta di qualcosina.

Vediamo per gradi cosa succede. Nelle righe 200-210 rimane sempre il nucleo di istruzioni da ripetere per ottenere un quadrato; in riga 220 però non troviamo l'istruzione LINEA che produce il "loop" ma l'istruzione TORNA. Se provate a far eseguire solo questo spezzone del programma riscontrerete un errore; E.LI.ANA. domanderà: "DOVE TORNO? ERRORE NELLA LINEA 220". In pratica un sottoprogramma non può essere eseguito così com'è, ma deve essere chiamato da un'altra istruzione alla quale appunto tornerà dopo essere stato eseguito.

L'istruzione di chiamata è naturalmente SOTTOPROGRAMMA; quando E.LI.ANA. trova questa istruzione salta alla riga specificata come primo parametro (nel nostro caso la 200) ed esegue fino a quando trova l'istruzione TORNA. Il secondo parametro invece indica il numero di ripetizioni dell'esecuzione del sottoprogramma; ed è proprio quest'ultimo che ci permette di costruire procedure cicliche eseguibili un numero finito di volte.

Nel caso del quadrato, la procedura in riga 200 che esegue un lato, se richiamata quattro volte, produce appunto un quadrato.

Analogamente, si può costruire un triangolo equilatero poichè l'angolo di rotazione e la lunghezza del lato sono uguali come nel quadrato; variano chiaramente i gradi ed il numero di lati. Questi sono tre così come indica il secondo parametro della chiamata SOTTOPROGRAMMA nella procedura qui di seguito:

[TRIANGOLO EQUILATERO]

```
100 ALBUM
110 CESTINO
120 SINISTRA 90
130 MATITA
140 SOTTOPROGRAMMA 200;3
150 FINE

200 DESTRA 120
210 AVANTI 30
220 TORNA
```

Il programma è analogo al precedente, l'unica differenza sostanziale consiste nella riga 120 che posiziona la tartaruga in modo che possa disegnare un triangolo con la punta verso l'alto. Questo accorgimento deve essere preso in grande considerazione, poichè sta alla base di tutti i lavori che utilizzano più volte la stessa procedura.

Ogni volta che si chiama una procedura, la tartaruga deve essere collocata nel punto in cui vogliamo che il disegno venga eseguito e con l'angolazione corretta. Per meglio comprendere il concetto, provate a spostare la tartaruga e variare la sua angolazione prima di chiamare uno dei sottoprogrammi che eseguono il quadrato o il triangolo.

Con l'istruzione sottoprogramma, quindi, possiamo scrivere procedure che saranno eseguite più volte e per un numero preciso di ripetizioni.

Prima di vedere come sfruttare più procedure all'interno di un programma, riteniamo utile aggiungere a quelle già viste una procedura rettangolo. In questo caso ci troviamo di fronte ad una soluzione leggermente diversa; abbiamo due coppie di lati di differente misura. La cosa si risolve scrivendo una procedura, richiamabile due volte, che esegue un lato "lungo" ed uno "corto".

[RETTANGOLO]

```
100 ALBUM
110 CESTINO
120 MATITA
130 SOTTOPROGRAMMA 200;2
140 FINE

200 AVANTI 20
210 DESTRA 90
220 AVANTI 40
230 DESTRA 90
240 TORNA
```

Ancora più sotto

Con questo "fardello" di sottoprogrammi (quadrato, triangolo, rettangolo) possiamo provare a costruire un programma più complesso. Torna utile riprendere la procedura "casa elettronica" del precedente numero, per affrontarla secondo questa nuova ottica. Dopo le solite righe preliminari (ALBUM, MATITA, CESTINO) potremmo cominciare la stesura scrivendo:

```
700 SOTTOPROGRAMMA 720;3
710 TORNA
720 DESTRA 120
730 AVANTI 30
740 TORNA

800 SOTTOPROGRAMMA 820;2
810 TORNA
820 AVANTI 20
830 AVANTI 30
835 DESTRA 90
840 TORNA
```

Queste righe di programma definiscono le procedure rettangolo e triangolo necessarie alla "costruzione" della "casa". Chiaramente, così come sono impostate non servono a nulla, nel senso che debbono essere richiamate a loro volta come sottoprogrammi; infatti le righe 710 e 810 contengono l'istruzione TORNA.

Una procedura "casa", che le utilizza, può essere scritta in questo modo:

```
500 SOTTOPROGRAMMA 800;1 [disegna il rettangolo]
505 AVANTI 20 [sposta la tartaruga in posizione corretta]
510 SINISTRA 90 [corregge l'angolazione]
515 SOTTOPROGRAMMA 700;1 [disegna il triangolo]
```

A questo punto la casa è "pronta", ma senza finestre; la procedura quadrato potrà ovviare a questo inconveniente se così consegnata:

```
600 SOTTOPROGRAMMA 620;4
610 TORNA
620 AVANTI 5
630 DESTRA 90
640 TORNA
```

Anche in questo caso bisogna provvedere alla necessaria chiamata ma, così come per le altre procedure, bisogna posizionare la tartaruga correttamente perchè le "finestre" siano collocate al posto giusto:

```
520 VOLA                    [posizionano]
525 DESTRA 90               [           ]
530 INDIETRO 5              [    la     ]
535 DESTRA 90               [           ]
540 AVANTI 5                [tartaruga]
545 MATITA                  [disegnano]
550 SOTTOPROGRAMMA 600;1
                            [la finestra]
```

Le righe dalla 520 alla 550 provvedono alla prima finestra; spostando avanti la tartaruga (riga 560) possiamo poi disegnare anche la seconda in posizione corretta:

```
555 VOLA
560 AVANTI 15
565 MATITA
570 SOTTOPROGRAMMA 600;1
575 TORNA
```

Scatole cinesi

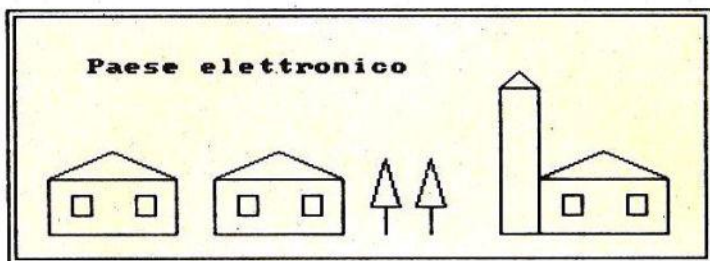
Ora abbiamo creato la procedura casa; possiamo contentarci di questo, oppure, in un secondo momento, possiamo decidere di diventare "imprenditori edili" lanciandoci nella costruzione di un intero paese. E' ovvio che in questa seconda impresa la procedura casa è destinata a diventare una piccola parte del "tutto". Basta continuare per la stessa strada, facendo diventare il programma "casa" sottoprogramma del programma "paese elettronico".

Viene da pensare alle scatole cinesi: un sottoprogramma che contiene un sottoprogramma che contiene un sottoprogramma... dove si finirà? Volendo, si potrebbe non finire mai (memoria permettendo) ma, generalmente, costruiti i vari sottoprogrammi, si organizza una procedura che li gestisca tutti; la procedura in questiona si chiama "programma principale" oppure "Main Program" e viene posta generalmente all'inizio del programma stesso.

Nel programma che questo mese vi proponiamo, "paese elettronico", tra la riga 100 e la 270 troviamo il "Main Program"; seguono le procedure "Chiesa", "Albero" e "Casa" che, chiamate diverse volte dal programma principale, provvedono, con opportuni posizionamenti della tartaruga, alla "costruzione" di questo simpatico "paese".

Dalla riga 600 alla 800 si susseguono le procedure di base, quelle che abbiamo usato per la "casa". queste ultime sono le procedure geometriche che forniscono gli elementi per la "costruzione".

L'esercizio migliore per approfondire i concetti esposti sin qui potrebbe consistere nell'apportare modifiche all'assetto urbanistico del paese; in parole povere sarebbe utile spostare le varie case, alberi ecc. aggiungendone magari qualcuna. Per fare questo bisogna tenere presente che per utilizzare correttamente la procedura casa, la tartaruga deve essere posizionata "a testa in su" e così anche per le procedure chiesa ed albero.

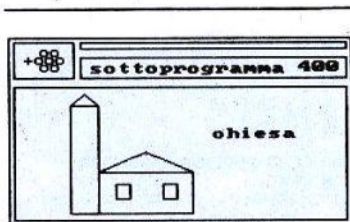
Chi volesse apportare modifiche utilizzando le sottoprocedure dalla riga 600 alla 800 deve servirsi dello stesso accorgimento tranne che per i triangoli, i quali devono essere richiamati posizionando la tartaruga ruotata a sinistra di 90 gradi rispetto al lato basso dello schermo.

Qualcuno, più esperto, potrebbe pensare che le cose qui proposte sono "troppo semplici"; effettivamente lo sono, se le guardiamo dal punto di vista dei risultati grafici ottenuti sul video. Ci preme quindi fare alcune precisazioni.

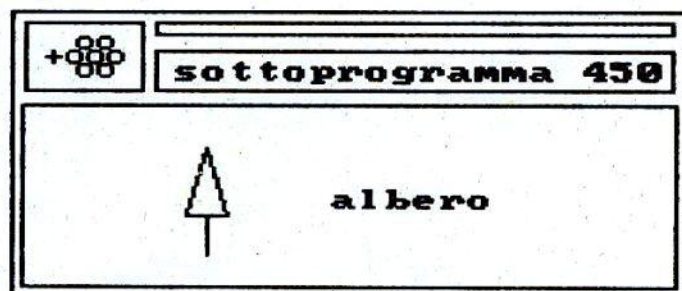
In queste righe stiamo cercando di realizzare una guida all'uso degli strumenti informatici attraverso un linguaggio estremamente semplice come E.LI.ANA. Conseguentemente, quello che maggiormente ci preme non è certo la realizzazione di "mirabolanti pirotecnie": fuochi artificiali se ne trovano in quantità sulle altre pagine della nostra rivista.

Moltissimi hanno sperimentato la confusione che nasce dalla costruzione di programmi troppo lunghi e male organizzati; questo è il problema di tutti: arrivati ad un certo punto non si riesce più a trovare la causa di alcuni errori, o, peggio, non ci si ricorda più come funzionava un determinato spezzone. Un buon programma si vede dalla bontà della sua struttura e non dalla lunghezza o dagli effetti che produce.

In questo trova fondamento la scelta che abbiamo fatto: iniziare con procedure semplici, per insegnare a strutturare un programma. Una volta assimilato questo modo di procedere si può intraprendere la strada dei "programmi pirotecnici", senza rischiare di incappare nei mali sopra accennati.

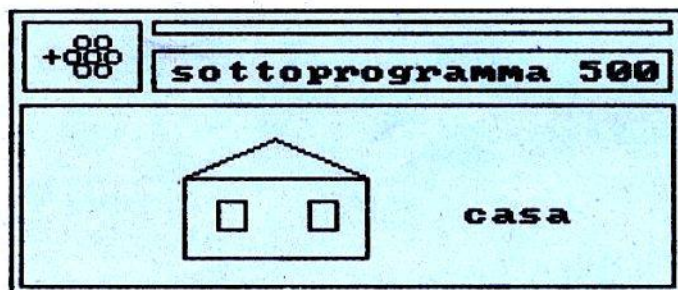
```
[PAESE ELETTRONICO]

100 ALBUM
105 CESTINO
110 VOLA
115 SINISTRA 90
120 AVANTI 150
125 DESTRA 90
130 INDIETRO 80
135 MATITA
140 SOTTOPROGRAMMA 500;1
145 VOLA
150 AVANTI 30
155 SINISTRA 90
160 INDIETRO 15
165 MATITA
170 SOTTOPROGRAMMA 500;1
175 VOLA
180 AVANTI 25
185 SINISTRA 90
190 INDIETRO 15
195 MATITA
200 SOTTOPROGRAMMA 450;1
205 VOLA
210 INDIETRO 30
215 DESTRA 90
220 INDIETRO 20
225 MATITA
230 SOTTOPROGRAMMA 450;1
235 VOLA
240 INDIETRO 50
245 DESTRA 90
250 INDIETRO 20
255 MATITA
260 SOTTOPROGRAMMA 400;1
270 LINEA 270
```

```
[ CHIESA ]

400 SOTTOPROGRAMMA 850;1
405 AVANTI 40
410 SINISTRA 90
415 SOTTOPROGRAMMA 750;1
420 INDIETRO 14
425 DESTRA 90
430 INDIETRO 40
435 SOTTOPROGRAMMA 500;1
440 TORNA
```

[ALBERO]

```
450 SOTTOPROGRAMMA 650;1
455 VOLA
460 SINISTRA 90
465 AVANTI 7
470 MATITA
475 SOTTOPROGRAMMA 750;1
480 TORNA
```

[CASA]

```
500 SOTTOPROGRAMMA 800;1
505 AVANTI 20
510 SINISTRA 90
515 SOTTOPROGRAMMA 700;1
520 VOLA
525 DESTRA 90
530 INDIETRO 5
535 DESTRA 90
540 AVANTI 5
545 MATITA
550 STTOPROGRAMMA 600;1
555 VOLA
560 AVANTI 15
565 MATITA
570 SOTTOPROGRAMMA 600;1
575 TORNA
```

[QUADRATO]

```
600 SOTTOPROGRAMMA 620;4
610 TORNA
620 AVANTI 5
630 DESTRA 90
640 TORNA
```

[LINEA]

```
650 AVANTI 20
660 TORNA
```

[TRIANGOLO 1]

```
700 SOTTOPROGRAMMA 720;3
710 TORNA
720 DESTRA 120
730 AVANTI 30
740 TORNA
```

[TRIANGOLO 2]

```
750 SOTTOPROGRAMMA 770;3
760 TORNA
770 DESTRA 120
780 AVANTI 14
790 TORNA
```

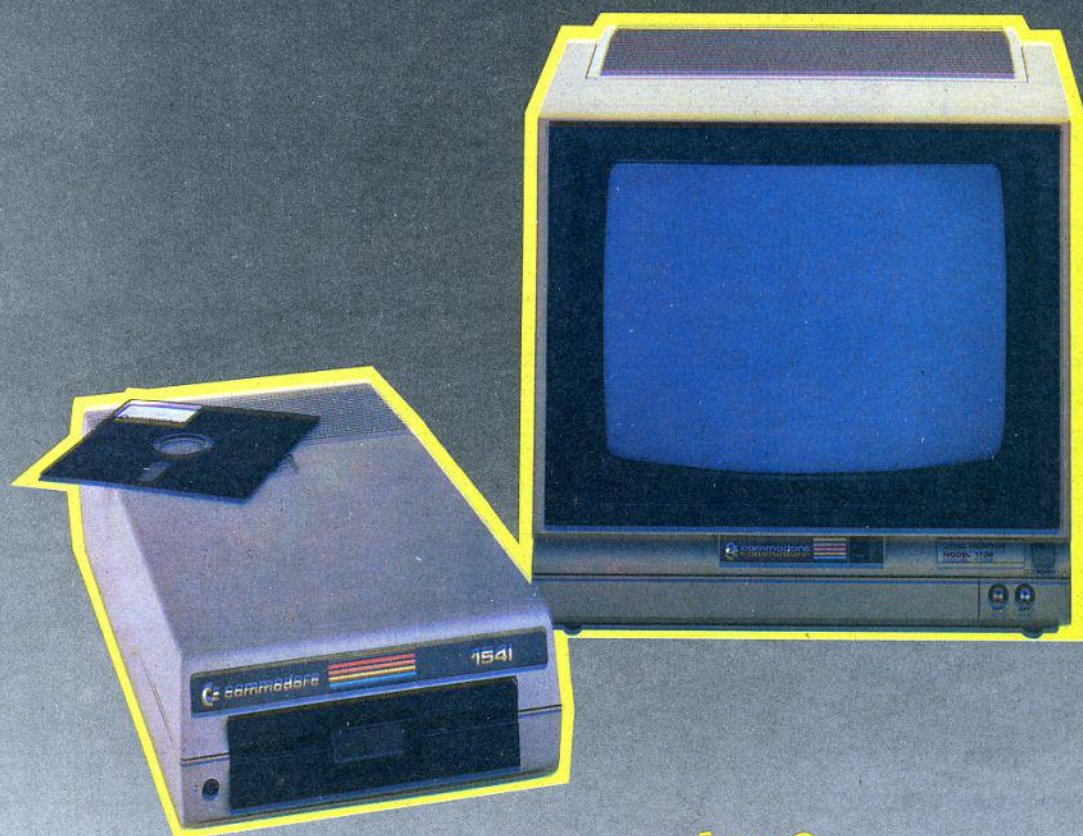
[RETTANGOLO 1]

```
800 SOTTOPROGRAMMA 820;2
810 TORNA
820 AVANTI 20
825 DESTRA 90
830 AVANTI 30
835 DESTRA 90
840 TORNA
```

[RETTANGOLO 2]

```
850 SOTTOPROGRAMMA 870;2
860 TORNA
870 AVANTI 40
875 DESTRA 90
880 AVANTI 14
885 DESTRA 90
890 TORNA
```

COMMODORE TI FA U

Prendi monitor o disk drive...

Grande Operazione "Fai di piú col tuo Commodore".

Monitor Commodore e disk drive 1541 sono le periferiche che ti permettono di utilizzare al meglio il tuo computer Commodore e di prepararti per il futuro.

Con il disk drive Commodore, memorizzi piú velocemente i dati su dischetti standard e puoi caricare i programmi in un attimo.

Con il monitor a colori ad alta definizione d'immagine puoi evidenziare le tue qualità di programmatore, utilizzando al massimo le capacità sonore e grafiche del tuo computer.

Monitor e disk drive oggi convengono, perchè hanno un prezzo eccezionale, come d'abitudine Commodore. Inoltre, proprio per dimostrarti quanto puoi fare di piú con queste periferiche, Commodore ti fa un regalo speciale.

Acquistando monitor o disk drive entro il

N REGALO SPECIALE

Enciclopedia delle routine

Seconda puntata

Come realizzare l'enciclopedia di routine ed utilizzarla nei propri listati.

A i lettori che hanno acquistato per la prima volta questo numero di Commodore Computer Club, illustriamo, in breve, i vantaggi che derivano dalla raccolta proposta:
- ogni routine appare, per intero, sullo schermo del computer e consente, proprio per questo motivo, di essere esaminata comodamente.
- ogni routine è numerata secondo uno standard che ha la particolarità di esser ricordato facilmente:

RIGHE	CONTENUTO
XXY00	Prima riga del sottoprogramma
XXY89	Ultima riga utile del sottoprogramma
XXY90	REM Prima riga di spiegazioni
XXY99	REM Nome della subroutine

in cui XX sono due valori variabili da 10 a 63; Y è un carattere numerico compreso tra 0 e 9.

Qualsiasi subroutine, in altre parole, inizia con un numero, di cinque caratteri, che termina sempre con "00". La stessa subroutine, d'altra parte, ha l'ultima riga numerata con "99".

Digitando, ad esempio:
LIST 10800-10899
si avrà la certezza di veder apparire sullo schermo, per intero, la routine il cui nome si trova nella riga 10899.

Prima di accedere alla routine è necessario assegnare, alle variabili indicate da riga XXY89 a XXY98, particolri valori per il suo corretto funzionamento. Al "ritorno" una o più variabili conterranno il risultato dell'elaborazione.

In questo modo, per esser più chiari, è possibile simulare alcuni comandi di versioni BASIC avanzate oppure, addirittura, creare nuove e inedite istruzioni.

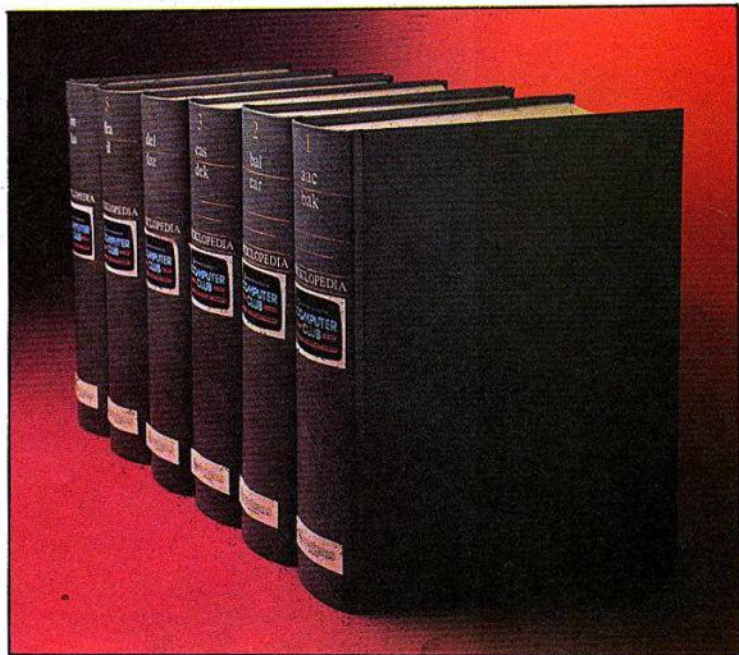
Ad esempio, il comando:
SOUND 1,800,500
che, nel C-16, riproduce un suono di tonalità 800 tramite la voce 1 per la durata 500, potrebbe venir riprodotta in un'ipotetica subroutine per il Commodre 64 con:

X1=1:X2=800:X3=500:GOSUB12400

nell'ipotesi, ovviamente, che la routine in oggetto sia allocata da riga 12400 a 12499.

La collaborazione dei lettori è gradita, purchè si provveda ad inviare almeno tre sottoprogrammi per volta, su nastro oppure su output di stampante. I listati di routine che non rispettano lo standard adottato non potranno esser presi in considerazione.

Tutti i lavori pubblicati verranno compensati con prodotti della Systems editoriale (cassette di programmi, libri, abbonamenti, copie arretrate eccetera).

Alessandro de Simone

10400 INCOLONNAMENTO VIRGOLA DECIMALE

Capita spesso di voler visualizzare su schermo un certo numero di valori che possono, oppure no, essere interi o decimali. Questa subroutine serve per incolonnarli nella giusta maniera.

Prima di accedere al sottoprogramma è necessario comunicare, oltre al valore desiderato (X1), anche la distanza dal bordo sinistro del video della prima cifra significativa (X2), che deve valere almeno 11 in modo da non perdere i primi caratteri nel caso di valori interi molto elevati.

Il semplice demo delle righe 100-150 genera, a gruppi di 10, valori Random che saranno incolonnati a distanza via via crescente. Al ritorno dalla subroutine è possibile recuperare anche la parte decimale (X4$) e quella intera (X3$).

N.B. Nella riga 10420 è necessario digitare il comando Return ricorrendo all'abbreviazione (reT).

```
  100 REM  ESEMPIO D'USO
  110 REM  INCOLON.VIRGOLA
  115 :
  120 X2=10
  130 FOR I=1 TO 10:X1=RND(0)*RND(0
      )*1E9
  140 GOSUB 10400:PRINTX1$CHR$(18)X
      3$CHR$(146);X4$:NEXT:PRINT:X2
      =X2+2:GOTO 130
  150 :
 9999 END
10400 X1$=STR$(X1):X3=LEN(X1$):IF X
      2<11 THEN X2=11
10403 IF X2>128 THEN X2=128
10405 X3$="":X4$="":X9$="":FOR X0=1
       TO X2:X9$=X9$+CHR$(32):NEXT
10410 FOR X0=1 TO X3:X2$=MID$(X1$,X
      0,1)
10420 IF X2$="." THEN X2$=RIGHT$(X9
      $,X2-X0):X1$=X2$+X1$:X4$=RIGH
      T$(X1$,X3-X0):RETURN
10430 X3$=X3$+X2$:NEXT
10440 X1$=LEFT$(X9$,X2-X3-1)+X1$:RE
      TURN
10489 REM  VARIABILI: X0,X1,X2,X3,X
      9,X1$,X2$,X3$,X4$,X9$
10490 REM  INPUT:X1$(VALORE) X2(DIS
      TANZA DAL BORDO SINISTRO)
10491 REM  OUTPUT: X1$=VALORE-STRIN
      GA NUMERO
10492 REM  X3$=PARTE INTERA
10493 REM  X4$=PARTE DECIMALE
10499 REM  NOME: INCOLONNAMENTO VIR
      GOLA
```

10500 INPUT CONTROLLATO

In molti programmi interattivi è necessario prevenire errori di battitura da parte dell'utilizzatore. La routine proposta, a seconda se la variabile X1 contenga il valore "0", "1" oppure "2", effettua un controllo sui tasti digitati in seguito.

Caso 0: accetta solo caratteri alfabetici (maiuscoli e minuscoli) e spazio (barra spaziatrice).

Caso 1: caratteri alfabetici (maiu./min.), spazio, caratteri numerici, segni (+-), punto decimale.

Caso 2: solo caratteri numerici, segni (+-), punto decimale. In questo caso viene effettuato un ulteriore controllo che non accetta il punto decimale nel caso sia stato già battuto, nè accetta un segno se questo non è il primo carattere digitato.

In tutti e tre i casi è possibile cancellare il carattere digitato per ultimo premendo l'apposito tasto Inst/del.

N.B. I lettori che abbiano difficoltà ad interpretare i caratteri di riga 10505 tengano conto che, dopo le virgolette (PRINT"), è necessario digitare i tasti che seguono: Reverse, spazio, cursore a sinistra, Off Reverse, spazio, Cursore a sinistra, chiusura virgolette, punto e virgola. Tale sequenza di caratteri simula il lampeggio del cursore.

```
  100 REM  ESEMPIO D'USO
  110 REM  INPUT CONTROLLATO
  115 :
  120 INPUT "TIPO DI INPUT:0,1,2";X
      1:GOSUB 10500
  130 PRINT:PRINT"[RVS]"X1$"[RVOFF]
      "LEN(X1$):GOTO 100
  200 :
 9999 END
10500 X1$="":X2=0
10505 GET X0$:IF X0$="" THEN PRINT"
      [RVS] [LEFT][RVOFF] [LEFT]";:
      GOTO 10505
10510 X0=ASC(X0$):IF X0=13 THEN RET
      URN
10515 IF X1 AND X0>47 AND X0<58 THE
      N 10580
10517 IF X1=2 AND X0=46 AND X2=0 TH
      EN X2=1:GOTO 10580
10518 IF X1=2 AND (X0=43 OR X0=45)
      AND LEN(X1$)=0 THEN 10580
10519 IF X1=1 AND (X0=46 OR X0=43 O
      R X0=45) THEN 10580
10520 IF X0=20 AND LEN(X1$) THEN 10
      580
10522 IF X1=2 THEN 10505
10525 IF X0=32 OR X0>64 AND X0<91 T
      HEN 10580
10530 IF X0>192 AND X0<218 THEN 105
      80
```

```
10570 GOTO 10505
10580 X1$=X1$+X0$:IF X0=20 AND LEN(
      X1$)>1 THEN X1$=LEFT$(X1$,LEN
      (X1$)-2)
10581 PRINTX0$;:GOTO 10505
10595 REM  X1=1 ALFANUMER. X1=0 SOL
      O ALFAB. X1=2 SOLO NUM. X1$=S
      TRINGA RITORNO
10599 REM  INPUT CONTROLLATO
```

10600 REVERSE SCHERMO

Questa routine è piuttosto interessante perchè consente di utilizzare un breve programma in linguaggio macchina (L.M) senza ricorrere alla consueta tecnica di READ... DATA... POKE...

I valori da pokare, infatti, sono contenuti nella stringa X0$ delle prime righe sotto forma di gruppi di tre valori. I primi codici (in decimale) sono, ad esempio: 173, 106, 3, 133, 2 eccetera. Un'apposita routine si incarica di "estrarre" i valori dalla stringa X0$, di valutarli e di pokarli nelle locazioni normalmente utilizzate dal registratore a cassette. La riga 10660 serve solo per controllare l'esatta trascrizione del listato da parte del lettore e deve essere cancellata dopo il controllo del suo corretto funzionamento.

La routine si occupa di cambiare, da normale in Reverse, tutti i caratteri presenti sullo schermo del Commodore 64, ad eccezione degli spazi bianchi, per un numero di volte specificato da X1 e con intervallo specificato da X2. Premendo un tasto durante il funzionamento della routine, si ottiene un ritorno al programma principale e la schermata ritorna come era prima della chiamata.

```
  100 REM  ESEMPIO D'USO
  110 REM  L.M. INVERSIONE
  115 REM  SCHERMO (SOLO C-64)
  120 :
  200 INPUT "N.VOLTE";X1
  210 INPUT "RITARDO";X2
  220 GOSUB 10600:GOTO 200
  225 :
10600 X1=INT(X1*2):IF X1<2 THEN X1=
      0
10640 X7=0:X8=0:X0$="17310600313300
      2173107003133003162000160000 1
      77002201032240"
10653 X0$=X0$+"004073128145002200 19
      2250208241024165002"
10656 X0$=X0$+"105250133002165003 10
      500013300323222400420822 10960
      00004"
10658 FOR X0=1 TO LEN(X0$) STEP 3:X
      9=VAL(MID$(X0$,X0,3)):POKE 82
      6+X8,X9:X8=X8+1
10659 X7=X7+X9:NEXT
10660 IF X7<>5332 THEN PRINT"ERRORE
      ":END
10662 REM  CANCELLARE RIGA PRECEDEN
      TE DOPO AVER VERIFICATO TRASC
      RIZIONE LISTATO
10664 REM  CANCELLARE ANCHE X7=0 E
      X7=X7+X9 IN RIGHE PRECEDENTI
10667 IF X1=0 THEN SYS826:RETURN
10670 FOR X0=1 TO X1:SYS826:FOR X9=
      1 TO X2
10675 GET X0$:IF X0$="" THEN 10685
10680 X0=X1:X9=X2:IF X0/2=INT(X0/2)
       THEN SYS826
10685 NEXTX9,X0:RETURN
10690 REM  X0,X1,X2,X7,X8,X9,X0$
10692 REM  INPUT X1(N.VOLTE PARI).
      X2(RITARDO REVERSE).SE X1<1 T
      HEN 1 VOLTA SOLA
10699 REM  REVERSE CARATTERI SCHERM
      O LM
```

10700 IMPULSI SONORI

Questa routine, valida solo per il Commodore 64, genera un certo numero (X2) di impulsi sonori separati da un intervallo di tempo proporzionale ad X1. Serve per richiamare l'attenzione dell'utilizzatore che, premendo un tasto, provoca il ritorno al programma principale.

Se il valore di X2 è compreso tra 1 e 100, si avrà un ritorno al programma anche senza la pressione di un tasto.

```
  100 REM  ESEMPIO D'USO
  110 REM  IMPULSI SONORI IN
  120 REM  ATTESA PRESSIONE TASTI
  130 :
  200 INPUT "DURATA IMPULS0";X1
  210 INPUT "N.VOLTE";X2
  220 GOSUB 10700:GOTO 200
  230 :
10700 X4=0:IF X1>1000 THEN X1=1000
10702 IF X2>100 OR X2<0 THEN 10710
10704 FOR X3=1 TO X2:GOSUB 10750
10706 GET X0$:IF X0$="" THEN NEXT
10708 RETURN
```

DAI MEMORIA AL TUO VIC-20

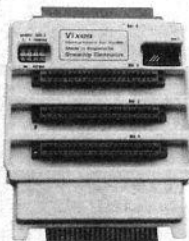
VIXEN SWITCHABLE 16K RAM

CON QUESTA CARTRIDGE POTRAI ESPANDERE LA MEMORIA DEL TUO VIC-20 COME VORRAI: 3K-8K-16K

Progettata e prodotta in Inghilterra dalla Stonechip Electronics questa cartuccia, tramite il semplice posizionamento di micro-switch, ti consente un ampliamento differenziato della **RAM: 3K o 8K o 16K,** secondo quanto richiede il programma che devi far girare. I **contatti placcati in oro,** assicurano una lunga durata a questa cartridge che darà nuova vita al tuo VIC-20.
Un prezzo introvabile e **garanzia per 6 mesi.**

PROGRAMMERS AID CARTRIDGE

Questa cartuccia aggiunge ben **19 comandi** al tuo **VIC-20:** renumber, autoline, edit, sound, cursor, block, delete .... e altri. E' un prodotto della Stonechip Electronics con **6 mesi di garanzia** integrale.

MODULO DI ESPANSIONE PER VIC-20

Questo modulo di espansione differenziata, prodotto dalla Stonechip Electronics, ti consente di **utilizzare contemporaneamente** (posizionando i microswitch di cui é dotato) da **2** a **4 cartridges.** Aggiunge inoltre alla **ROM 4K.** Inserendo con l'espansione Vixen 16K Ram una espansione standard di 8K si ottengono ben **24K RAM.** E' anche dotato di una presa per collegare un ulteriore modulo di espansione. Garanzia 6 mesi.

SENSAZIONALE PREZZO PER CARTRIDGES COMMODORE VIC-20

Scegli i videogiochi che preferisci a sole 7.350 Lire + iva
1901 Avenger - 1907 Jupiter lander - 1910 Rat race - 1912 Mole attack - 1913 Ride on Fort Knox - 1914 Adventureland 1915 Pirate cove - 1918 Voodoo castle - 1919 Sargon chess II - 1922 Cosmich cruncher - 1923 Gorf - 1924 Omega race

PER IL TUO COMMODORE-16 UN'ESPANSIONE RAM DI 16K

La consueta qualità e affidabilità della Stonechip Electronics per questa cartridge, che potenzia il tuo C - 16. Garanzia integrale di sei mesi.

INTERFACCIA PER REGISTRATORE PER VIC-20 E PER CBM-64

La Stonechip Electronics é stata la prima ditta in Europa a realizzare l'interfaccia per utilizzare qualsiasi registratore con i computer Commodore. Continuamente aggiornata é di certo una delle più affidabili presenti sul mercato. Garanzia di 6 mesi.

12 CASSETTE C-20 A LIRE 12.600 + IVA
Nastro professionale AGFA PE619, box trasparente in Kofil assemblato con 5 viti, scorrimento su perni in acciaio lubrificati.

50 FANTASTICI VIDEOGIOCHI 50 PER VIC-20 O CBM-64

Registrati su nastro, della software-house inglese Cascade Games, all'**inverosimile prezzo** di sole **Lire 25.000 + IVA !!**

Spedire il presente **MODULO D'ORDINE,** o fotocopia, in una busta chiusa, unendo L. 2.600 in francobolli per spese postali.
UN OMAGGIO A CHI ORDINA ALMENO DUE ARTICOLI

.... ✂

Spett.le **APCO s.r.l. - Cas. Post. 239 - 10015 IVREA (To)**
desidero ricevere quanto da me contrassegnato con X. Pagherò direttamente al postino gli importi qui elencati, che sono comprensivi di IVA e di spese di imballo e contrassegno.

- ☐ Vixen Switchable 16K RAM — a L. 87.000
- ☐ Programmer Aid Cartridge — a L. 48.000
- ☐ Modulo di espansione per VIC-20 — a L. 75.000
- ☐ Cartridges VIC-20 cod. .... cadauna a L. 9.000
- ☐ 12 computer cassette C-20 — a L. 16.000
- ☐ Interfaccia per registratore — a L. 31.000
- ☐ Espansione di memoria per Commodore-16 — a L. 89.000
- ☐ 50 videogiochi per ☐ VIC-20 ☐ CBM-64 — a L. 30.000
- ☐ Non acquisto nulla ma desidero ricevere i Vostri cataloghi

Nome e Cognome ______________________________

Via ______________________________

Città ______________________________ CAP. ________

data ________________ firma ________________

```
10710 GET X0$:IF X0$="" THEN GOSUB
      10750:GOTO 10710
10720 RETURN
10750 POKE 54296,15:POKE 54277,9:PO
      KE 54278,12
10760 POKE 54272,49:POKE 54273,28:P
      OKE 54276,17
10770 FOR X0=1 TO X1:NEXT:POKE 5427
      6,0:RETURN
10790 REM  X0,X1,X2,X3,X0$
10792 REM  X1<1000 DURATA IMPULSO
10794 REM  X2= N.IMPULSI (0-100)
10796 REM  X2<0 ATTESA PRESSIONE TA
      STO
10799 REM  IMPULSI SONORI IN ATTESA
```

10800 CONTROLLO DELLA DATA

Questo interessante e breve sottoprogramma consente di effettuare diversi controlli a seconda del contenuto di alcune variabili:

X4=0. Verifica se un certo giorno "esiste" oppure no. Rispondendo, ad esempio, al breve programma dimostrativo di righe 100-290:

GIORNO? 12
MESE? 22
ANNO? 85

la variabile X0$ conterrà il messaggio "ERR1" che indicherà un errore di INPUT in una delle tre variabili (X1, X2, X3). Non esiste, infatti, il mese N.22.

Il controllo viene effettuato, oltre che sul mese, sul numero massimo di giorni per ciascun mese e sulla esistenza, o meno, del 29 febbraio degli anni bisestili.

L'anno può esser digitato in molti modi. Esempio: il 1985 può esser indicato come 1985, 985 oppure, semplicemente, 85.

Se la data "esiste", al ritorno della subroutine la variabile X3$ sarà nulla. Se, invece, contiene i caratteri "ERR2" indicherà che alla varibile X1$ non è associato un corretto nome della settimana (LUN, MAR, eccetera).

Se, prima di utilizzare la subroutine, ad X4 viene assegnato un valore maggiore di zero, e se X1$ contiene un nome lecito di giorno, viene effettuata una somma allo scopo di sapere il giorno che capiterà dopo "X4" giorni.

Esempio:
GIORNO? 14
MESE? 9
ANNO? 1985
N.GIORNI DA SOMMARE? 67
NOME GIORNO? SAB

La risposta che si ottiene è:
20 11 1985 MER

Ciò significa che se il giorno 14/9/85 è un sabato, dopo 67 giorni la data esatta sarà: mercoledì 20/11/85

Eliminando dalla riga 10826 lo statement REM sarà possibile visualizzare tutti i giorni calcolati durante la procedura.

```
  100 REM  ESEMPIO D'USO
  110 REM  CONTROLLO DELLA
  111 REM  DATA (G/M/A/NOME)
  115 :
  200 INPUT "GIORNO";X1
  210 INPUT "MESE";X2
  220 INPUT "ANNO";X3
  230 INPUT "N.GIORNI DA SOMMARE";X
      4
  240 INPUT "NOME GIORNO (LUN MAR..
      .)";X1$
  250 GOSUB 10800:PRINTX0$;X3$:PRIN
      TX1;X2;X3;X1$:GOTO 200
  290 :
 9999 END
10800 X2$="LUNMARMERGIOVENSABDOM"
10801 GOSUB 10851:GOSUB 10830:IF X0
      $="ERR1" OR X3$="ERR2" THEN R
      ETURN
10803 IF X4<1 THEN RETURN
10805 FOR X5=0 TO X4-1:X1=X1+1:GOSU
      B 10830
10815 IF X0$="ERR1" THEN X1=1:X2=X2
      +1:IF X2>12 THEN X2=1:X3=X3+1
10820 X6=X6+3:IF X6>21 THEN X6=1
10826 X1$=MID$(X2$,X6,3)
10827 REM  PRINTX1;X2;X3;X1$
10828 NEXT:X0$="":RETURN
10830 X0$="":X0=X1*X2*(X2<13)*(X1<3
      1+((X2+(X2>7)) AND 1)+(X2=2)*
      (2+((X3 AND 3)=0)))
10840 IF X0=0 THEN X0$="ERR1"
10850 RETURN
10851 X3$="ERR2":FOR X5=1 TO 21 STE
      P 3:IF MID$(X2$,X5,3)=X1$ THE
      N X6=X5:X5=22:X3$=""
10855 NEXT:RETURN
10890 REM  X0,1,2,3,4,5,6,X0$,1$,2$
      ,3$
10892 REM  X1,X2,X3 INPUT GIO,MES,A
      NNO
10894 REM  X4,X1$ INPUT N.GIORNI,NO
      ME G.
10896 REM  OUTPUT:X1,2,3,X1$,0$,3$(
      ERR.)
10899 REM  CONTROLLO DATA E SOMMA G
      IORNI
```

Un suggerimento

Un lettore ha consigliato di utilizzare le ultime righe del BASIC come indice generale delle routine costituenti l'intera enciclopedia. L'idea è decisamente interessante e la utilizziamo per indicare non solo le routine di questo numero ma anche quelle apparse sul numero precedente di Commodore Computer Club.

```
63989 REM 10800 CONTROLLO DATA
63990 REM 10700 IMPULSI SONORI
63991 REM 10600 REVERSE SCHERMO
63992 REM 10500 INPUT CONTROLLATO
63993 REM 10400 INCOLONNAMENTO VIRGOLA
63995 REM 50000 LEGGE BLOCKS LIBERI (DISCO)
63996 REM 10300 INPUT CON CONTROLLO DI DEFAULT
63997 REM 10200 ESTRAZIONE PAROLA DA FRASE
63998 REM 10100 CAMBIA COLORE BORDO & FONDO
63999 REM 10000 CORNICE POLICROMA
```

Alessandro de Simone

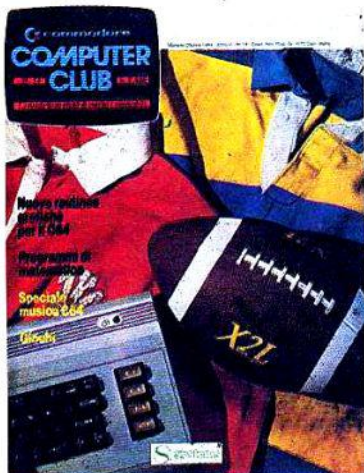
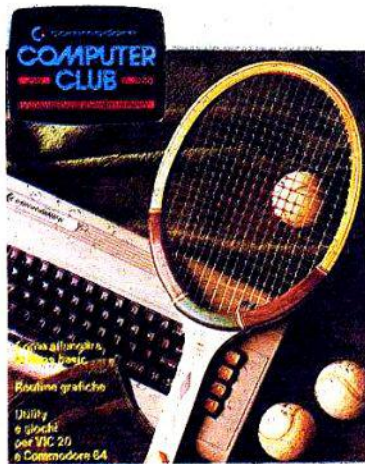
Il N.10

Il numero di pagine passa da 52 a 68 mentre il prezzo di copertina (2500 lire) rimane invariato e rimarrà tale fino al N.15 (Nov.84).

Tra gli argomenti di questo fascicolo: gestione dei puntatori vitali per i sistemi Commodore, utility di programmazione, videogiochi, un... pesce d'aprile e, soprattutto, un programma per disegnare in alta risoluzione sul Commodore 64: "E' solo la prima realise di un programma più impegnativo che sto portando avanti" afferma Danilo Toma, che promette di pubblicare, in seguito, qualcosa di unico nel suo genere.

Il N.7

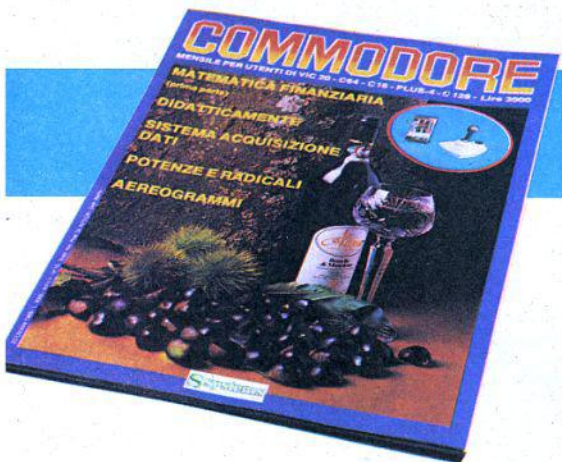
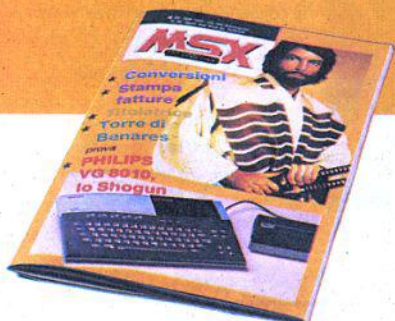
Il Commodore 64 è ora venduto in migliaia di esemplari ed i suoi utilizzatori iniziano a richieder programmi. Commodore Computer Club si presenta in edicola, col N.7, proponendo un programma universale di Assembler, listati per l'astro nascente C-64, una maschera - memo da ritagliare ed applicare sul proprio computer (Vic 20 oppure C-64), la prima puntata sull'Assembler.

Per la prima volta in Italia una rivista di informatica è costretta alla ristampa dopo una settimana di presenza in edicola. La notizia, ovviamente, si diffonde tra le case editrici ed inizia l'invasione di riviste e rivistine dedicate ai prodotti Commodore. Gli scettici dell'anno precedente hanno cambiato bandiera, seguendo il vento.

Il N. 14

E' il numero più richiesto dai nuovi lettori. Il motivo è semplice: sono pubblicate le nuove routine grafiche che consentono, aggiundengo nuovi comandi Basic, di disegnare secondo le regole della prospettiva. Nessun programma, su piccoli computer, è in grado di sviluppare una grafica di tale livello.

Il numero di pagine passa da 68 a 84. Il prezzo di 2500 lire, decisamente modesto, potrà restare invariato per un solo numero ancora.

L'UNIONE F

SINCLAIR COM

COMMODORE

MSX

A LA FORZA.

PUTER

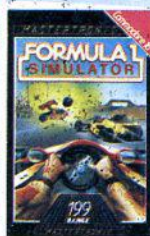
DA GENNAIO in edicola

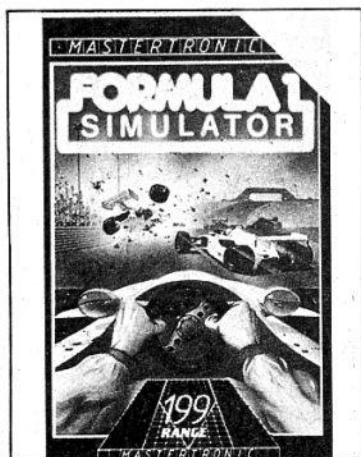
Tre riviste in una: tutto quello che conoscete già con qualcosa in piu'. Le macchine: nuovi prodotti, accessori, prezzi, confronti. Il software: tanto software, in piu' versioni per piu' computer. Le idee: tanto spazio per le vostre idee (e per le nostre). Da gennaio. In edicola.

Il mercato si evolve. Anche noi.

Il N. 20

Per venire incontro ai principianti vengono pubblicati, oltre a listati di un certo "livello", soprattutto articoli e mini programmi di grande aiuto per chi ha appena acquistato un com-

puter e desidera utilizzarlo realmente. L'articolo su come realizzare videogiochi, l'introduzione alla gestione degli archivi su disco, la scelta del modulo opportuno nei progetti di ruote dentate, non rappresentano che alcuni degli articoli didattici che troveranno ampio consenso tra i nostri lettori.

giochiamo con....

a cura di **Armon**

The empire

Tre rovinose guerre hanno ridotto l'impero di Artaivius alla rovina. Una piccola colonia di ribelli si è impadronita di una vecchia astronave e ha deciso di rendere ancora più disastrosa la cruda realtà di Artavius. I ribelli hanno piazzato cinque super mine, e il tuo compito è quello di disattivarle. A rendervi il lavoro ancora più complicato sarà la vostra energia vitale che diminuisce a vista d'occhio. Per disinnescare queste mine dovrete entrarvi dentro. Arrivare fino al nucleo. Disattivarlo, ed infine uscire. Tutte queste operazioni dovranno essere eseguite senza farvi scoprire dai robot sentinella che si trovano in tutti i corridoi all'interno delle mine.

Questa azione dovrete ripeterla 4 volte questo perchè solo quando saranno rese inattive le mine si potrà avere accesso alla quinta. Quest'ultima infatti è l'obiettivo finale della partita.

Quando viaggiate nello spazio con la vostra astronave che si chiama AIR WOLF 2000 fate attenzione a non schiantarvi contro le meteore che vi potranno distruggere.

Tasti funzione:

M = destra
N = sinistra
V = selezione missione
I = fuoco
Q = alto
A = basso

Consigli:

1) Tenetevi al centro dello schermo.
2) Entrando in una mina è necessario usare lo schermo principale per meglio riuscire allo scopo.
3) Evitare il contatto con i robot all'interno dei corridoi delle mine altrimenti si perde la preziosa energia in maniera veloce.

IDEA	7
GIOCABILITA'	7
SUONO	8
ANIMAZIONE	8
VOTO	8

Formula 1 simulator

Se non avete mai potuto chiedere fino ad ora a un computer di potervi trasportare in una macchina per farvi provare il brivido dei 300 K/h, tenuta di strada eccezionale e con dei buonissimi freni, questo è il vostro gioco. Siete voi l'abile pilota di una delle Ferrari e dovrete qualificarvi nel miglior posto. Allora via, ingranate la marcia, curvate nel momento giusto e superate le auto degli avversari.

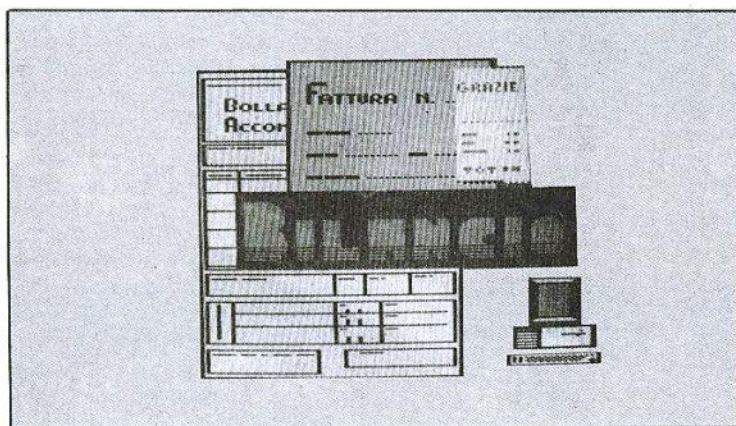
Ogni giro il punteggio vi aumenterà.

Gareggiando contro voi stessi vi potrete esercitare per ottenere un punteggio sempre più alto. Se giocate invece, contro i vostri amici, allora, il punteggio dovrà essere superiore a quello dei vostri amici.

Tasti comando:

A = accelerazione
Z = decelerazione
3 = sinistra
4 = destra
T = marce

IDEA	9
GIOCABILITA'	7
SUONO	8
ANIMAZIONE	7
VOTO	8

SPAZIO COMMODORE CLUB

Il bilancio di casa

Più che uno speciale, il prodotto reperibile in edicola, è un validissimo strumento utile per le esigenze di famiglia.

Realizzato da un professionista della programmazione per la Systems editoriale, viene sottoposto all'attenzione dei nostri lettori.

Finalmente una buona idea per cominciare ad utilizzare il nostro Commodore 64 anche per mettere un pò di ordine nei conti domestici.

Il programma che vi presentiamo nel numero speciale di Commodore Computer Club va altrettanto bene sia per gestire situazioni complesse come il bilancio di tutta una famiglia, sia per situazioni meno impegnative ma non per questo semplici come il bilancio di un giovane.

Questa elasticità d'uso è garantita dal fatto che sarete voi a comunicare al programma la descrizione delle spese e delle entrate che volete seguire: il programma si limiterà (si fa per dire) a fare bene le sue somme.

In realtà il programma fa ben altro, in quanto vi fornirà, oltre ai totali mensili ed annuali dei vostri conti, anche dati statistici in percentuale ed in assoluto delle vostre spese ed entrate.

E' prevista la possibilità di confrontare una particolare voce di spesa con le stesse dei mesi precedenti ed in tutto l'anno in corso, nonchè la possibilità di ottenere tutti questi dati anche sotto forma di grafici.

Il programma funziona, oltre che con il registratore a cassette, anche con l'unità a disco 1541 ed i possessori di una stampante potranno ottenere su carta i grafici, i dati statistici ed i vari totali e saldi che il programma vi mette a disposizione. La stampante ed il disco non sono tuttavia strettamente necessari per il regolare funzionamento del programma.

Quanto alla necessità di avere una gestione "meccanizzata" delle proprie uscite ed entrate, molti sono i punti a favore.

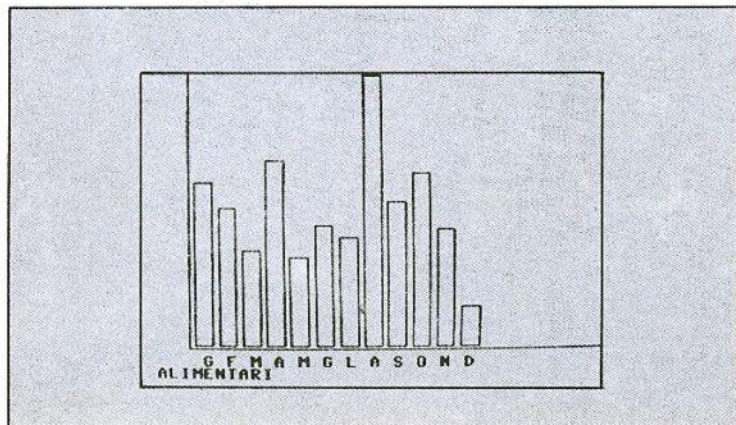
Per prima cosa bisogna riconoscere che, a meno di avere entrate ed uscite così esigue da riuscire a gestirsele a memoria, (cosa che non vi auguro di certo) un movimento "monetario" medio richiede una certa cura per essere seguito.

Si tratta il più delle volte di riuscire a sapere in anticipo se determinate spese importanti potranno essere sostenute o se dovremo rinviare a tempi migliori i nostri desideri.

Sia che si tratti di una macchina o di una casa per una famiglia, sia che si tratti di un motorino, di un disk drive 1541 o dei risparmi per le vacanze di un giovane studente, la esigenza è sicuramente quella di sapere cosa si risparmia mediamente ogni mese e di sapere eventualmente quali possano essere le spese su cui intervenire per risparmiare di più.

Tutto ciò è notevolmente semplice se usate un elaboratore ed un programma fatto espressamente per la valutazione dei dati di un bilancio domestico.

Come seconda considerazione si può dire che, poichè abbiamo un Commodore 64 in casa, tanto vale usarlo, non solo per i videogiochi e per qualche rubrica telefonica (argomenti certo di tutto rispetto) ma anche per un utilizzo diverso che indirettamente potrà dare anche soddisfazioni economiche.

Sul numero speciale di novembre troverete una guida completa all'uso del programma.

Alcuni utili consigli e trucchi per i lettori di Commodore Computer Club.

I listati della Systems Editoriale

Abbiamo notato che chi ha appena acquistato un computer Commodore si trova di fronte ad alcuni dubbi apparentemente irrisolvibili. Per venire incontro ai neo-possessori, riportiamo qui di seguito alcuni consigli indubbiamente preziosi:

1 - Se utilizzate il registratore è buona norma, subito dopo il salvataggio (=registratore) dei programmi che digitate da riviste, effettuare l'operazione di verifica. Questa si compie riavvolgendo il nastro e digitando il comando VERIFY seguito dalla pressione del tasto Return (vedi libretto di istruzioni allegato al computer). Se alla fine dell'operazione compare il messaggio VERIFY ERROR, è necessario ripetere l'intera procedura (registrazione + verifica).

2 - Se il caricamento dei programmi (da nastro o disco) avviene con difficoltà (LOADING ERROR), è probabile che ciò sia causato dalla vicinanza degli apparecchi al televisore. Allontanandoli tra loro si impedisce ai campi magnetici di alcuni TV di influire negativamente sul trasferimento dei dati.

3 - Se l'allontanamento del registratore dall'apparecchio TV non produce alcun vantaggio, è probabile che il programma sia stato registrato male o che la testina del registratore sia disallineata. Con molta cautela, in quest'ultimo caso, ruotare con apposito cacciavite la vite di regolazione di *non più* di un quarto di giro in verso orario oppure antiorario fino a che il caricamento non riesce.

4 - I programmi che è possibile trascrivere sul vostro calcolatore sono ben specificati. I listati "universali" riportano, su Commodore Computer Club, la dicitura "QUALSIASI COMPUTER". E' possibile, con le dovute eccezioni, adattare listati specifici per il Commodore 64 al Vic 20, al C-16 o viceversa. Questa è però un'operazione che può riuscire a persone piuttosto esperte e che possono disporre dei vari computer: non esiste, infatti, una regola unica e generale di adattamento di programmi ma, volta per volta, le regole da applicare cambiano di continuo e necessitano in ogni caso di una verifica sul calcolatore.

5 - Un'elevata percentuale dei nostri lettori incontra difficoltà nel digitare i programmi da noi pubblicati. Riportiamo, pertanto, alcune informazioni utili per la digitazione dei listati: i caratteri "speciali" bianchi su fondo nero (semi-grafici in "reverse") che rappresentano precisi comandi per i computer Commodore, sono riportati nel listato di esempio nella pagi-

na accanto così come appaiono digitandoli su video o su stampante, mentre a destra come li rappresentiamo nei nostri listati.

La riga 360, ad esempio, deve così esser interpretata:

Dopo aver battuto il carattere di virgolette (") che si ottiene premendo il tasto SHIFT insieme con il tasto 2, è necessario battere il carattere CRSR DOWN (il tasto, cioè, che normalmente sposterebbe il cursore nella cella video sottostante).

Analogamente, nella riga 180 del listato "tradotto" (di destra), il termine [NERO] sta a significare che occorre utilizzare il carattere speciale del colore nero (tasto CTRL insieme con il tasto 1, vedi listato).

Per ricordare in che modo vengono normalmente visualizzati i caratteri speciali, nella seconda parte di ogni riga (dopo i REM) sono riportati i tasti che è necessario premere per ottenere il carattere-comando "speciale".

La Redazione

```
100 REM I CARATTERI SPECIALI
110 REM DEI COMPUTER COMMODORE
120 REM COME APPAIONO NORMALMENTE
130 REM SU VIDEO O SU CARTA.
140 REM (CTRL = TASTO CONTROL)
150 REM (CMDR = TASTO COMMODORE)
160 REM (CRSR = TASTI CURSORE)
170 :
180 PRINT"■":REM CTRL+1 NERO
190 PRINT"■":REM  "  +2 BIANCO
200 PRINT"■":REM  "  +3 ROSSO
210 PRINT"◣":REM  "  +4 AZZURRO
220 PRINT"■":REM  "  +5 PORPORA
230 PRINT"■":REM  "  +6 VERDE
240 PRINT"■":REM  "  +7 BLU
250 PRINT"■":REM  "  +8 GIALLO
260 PRINT"■":REM  "  +9 REVERSE ON
270 PRINT"■":REM  "  +0 REVERSE OFF
280 PRINT"■":REM CMDR+1 ARANCIO
290 PRINT"■":REM  "  +2 MARRONE
300 PRINT"■":REM  "  +3 ROSSO CHIARO
310 PRINT"■":REM  "  +4 GRIGIO 1
320 PRINT"■":REM  "  +5 GRIGIO 2
330 PRINT"■":REM  "  +6 VERDE CHIARO
340 PRINT"■":REM  "  +7 BLU CHIARO
350 PRINT"■":REM  "  +8 GRIGIO 3
360 PRINT"■":REM  CRSR IN BASSO
370 PRINT"■":REM  CRSR A DESTRA
380 PRINT"□":REM  CRSR IN ALTO
390 PRINT"■":REM  CRSR SINISTRA
400 PRINT"■":REM  HOME
410 PRINT"■":REM  CANCELLA SCHERMO
420 :
430 REM  ESEMPI DI VISUALIZZAZIONE:
440 PRINT"■■■■■■■":REM CANCELLA SCHERMO,
450 :                REM CRSR DWN DUE VOLTE
460 :                REM CRSR DESTRA TRE "
470 :
480 PRINT"■■■■":REM BIANCO,CRSR SINISTRA
490 :            :REM DUE VOLTE E CRSR DWN
500 :            :REM UNA SOLA VOLTA
```

```
100 REM  I CARATTERI
110 REM  SPECIALI: COME
120 REM  VENGONO INDICATI
130 REM  SULLE RIVISTE:
140 REM    COMMODORE
150 REM  E COMMODORE
160 REM  COMPUTER CLUB.
170 :
180 PRINT"[NERO]"
190 PRINT"[BIANCO]"
200 PRINT"[ROSSO]"
210 PRINT"[AZZUR]"
220 PRINT"[VIOLA]"
230 PRINT"[VERDE]"
240 PRINT"[BLEU]"
250 PRINT"[GIALLO]"
260 PRINT"[RVS]"
270 PRINT"[RVOFF]"
280 PRINT"[ARANC]"
290 PRINT"[MARR]"
300 PRINT"[ROSA]"
310 PRINT"[GRIGIO1]"
320 PRINT"[GRIGIO2]"
330 PRINT"[VERDE2]"
340 PRINT"[CELESTE]"
350 PRINT"[GRIGIO3]"
360 PRINT"[DOWN]"
370 PRINT"[RIGHT]"
380 PRINT"[UP]"
390 PRINT"[LEFT]"
400 PRINT"[HOME]"
410 PRINT"[CLEAR]"
420 :
430 REM   ESEMPI
440 PRINT"[CLEAR][2 DOWN]
    [4 RIGHT]"
450 :
460 :
470 :
480 PRINT"[BIANCO][2 LEFT
    ][DOWN]"
```

INVIARE TUTTA LA PAGINA ANCHE SE SI UTILIZZA UNA SOLA SCHEDA

Nome .... **Cognome** ....

Via .... **N°** .... **CAP.** .... **Città** ....

Telefono .... **Orario** ....

Registrate il mio abbonamento annuale a Commodore Computer Club.

☐ Ho versato oggi stesso il canone di L. 28.000 a mezzo c/c postale n° 37952207 intestato a:
Systems Editoriale - V.le Famagosta, 75 - 20142 Milano

☐ Ho inviato oggi stesso assegno bancario n ....
per l'importo di L. 28.000 intestato a Systems Editoriale

Si prega di scrivere il proprio nome e l'indirizzo completo in modo chiaro e leggibile. Inviare la fotocopia del bollettino di c/c postale.

Considerando che i numeri 1, 2 e 7 sono esauriti, vogliate inviarmi i numeri arretrati .... al prezzo di L. 5.000 cadauno per richieste fino a 4 numeri, o di L. 4.000 cadauno per richieste oltre i 4 numeri arretrati, e perciò per un totale di L.............. Sono a conoscenza che i fascioli suddetti non saranno inviati in contrassegno e, pertanto, ho provveduto oggi stesso a versare il canone di L.................. a mezzo c/c postale n. 37952207 intestato a:
Systems Editoriale - V.le Famagosta, 75 - 20142 Milano

STATISTICA

Non posseggo un computer ☐

Posseggo un C64 .... si ☐ ... no ☐
Posseggo un VIC 20 .... si ☐ ... no ☐
Posseggo un Commodore Plus 14 .... si ☐ ... no ☐
Posseggo un Commodore Plus 16 .... si ☐ ... no ☐
Posseggo un registratore dedicato .... si ☐ ... no ☐
Posseggo un drive 1541 .... si ☐ ... no ☐
Posseggo una stampante .... si ☐ ... no ☐
Posseggo un monitor .... si ☐ ... no ☐

COLLABORAZIONE

A titolo di prova vi invio un articolo e la cassetta........................disco..............................
col programma che intendo proporre per la pubblicazione di cui garantisco l'originalità.

DOMANDA/RISPOSTA ....

....

....

....

....

....

RICHIESTA ARGOMENTI

Mi farebbe piacere che Commodore Computer Club parlasse più spesso dei seguenti argomenti:

1/ ....

2/ ....

3/ ....

4/ ....

GIUDIZIO SUI PROGRAMMI DI QUESTO NUMERO

Ho assegnato un voto da 0 a 10 ai programmi che indico di seguito:

A/ .... Voto ....

B/ .... Voto ....

C/ .... Voto ....

D/ .... Voto ....

PICCOLI ANNUNCI ....

....

....

....

....

....

CERCO/OFFRO CONSULENZA ....

....

....

....

....

....

INVIARE IN BUSTA CHIUSA E AFFRANCANDO SECONDO LE TARIFFE VIGENTI A:

COMMODORE COMPUTER CLUB

V.le Famagosta, 75
20142 Milano

INVIARE TUTTA LA PAGINA ANCHE SE SI UTILIZZA UNA SOLA SCHEDA

Nome ____________ **Cognome** ____________

Via ____________ **N°** ______ **CAP.** ______ **Città** ____________

Telefono ____________ **Orario** ____________